(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 February 2005 (24.02.2005)

PCT

(10) International Publication Number
WO 2005/017468 A2

(51) International Patent Classification[7]: G01F 1/20

(21) International Application Number:
PCT/US2004/026444

(22) International Filing Date: 13 August 2004 (13.08.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/495,447 15 August 2003 (15.08.2003) US
10/917,719 12 August 2004 (12.08.2004) US

(71) Applicant *(for all designated States except US)*: **APPLE COMPUTER, INC.** [US/US]; 1 Infinite Loop, MS 3-PAT, Cupertino, CA 95014 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CULBERT, Michael** [US/US]; 18500 Hillview Drive, Monte Sereno, CA 95030 (US). **COX, Keith, Alan** [US/US]; 1234 Colleen Way, Campbell, CA 95008 (US). **HOWARD, Brian** [US/US]; 59 Linaria Way, Menlo Park, CA 94028 (US). **DE CESARE, Josh** [US/US]; 625 Lisa Way, Campbell, CA 95008 (US). **WILLIAMS, Richard, Charles** [US/US]; 13198 Via Madronas Drive, Saratoga, CA 95070 (US). **FALKENBURG, Dave, Robbins** [US/US]; 5199 Bela Drive, San Jose, CA 95129 (US). **HUANG, Daisie, Iris** [US/US]; 5187 Saddle Brook Drive, Oakland, CA 94619 (US). **RADCLIFFE, Dave** [US/US]; 910 Exmoor Way, Sunnyvale, CA 94087 (US).

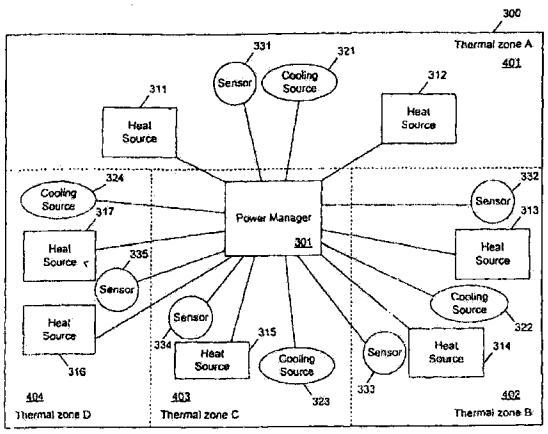
(74) Agents: **SCHELLER, James, C., Jr.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: METHODS AND APPARATUS FOR OPERATING A DATA PROCESSING SYSTEM



(57) Abstract: Methods and apparatuses to manage working states of a data processing system. At least one embodiment of the present invention includes a data processing system with one or more sensors (e.g., physical sensors such as tachometer and thermistors, and logical sensors such as CPU load) for fine grain control of one or more components (e.g., processor, fan, hard drive, optical drive) of the system for working conditions that balance various goals (e.g., user preferences, performance, power consumption, thermal constraints, acoustic noise). In one example, the clock frequency and core voltage for a processor are actively managed to balance performance and power consumption (heat generation) without a significant latency. In one example, the speed of a cooling fan is actively managed to balance cooling effort and noise (and/or power consumption).

WO 2005/017468 A2

ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND APPARATUSES FOR OPERATING A DATA PROCESSING SYSTEM

[0001] This application is related to and claims the benefit of the filing date of U.S. provisional application serial no. 60/495,447, filed August 15, 2003, and entitled "Methods and Apparatuses for Operating a Data Processing System" by the inventors Michael Culbert, Keith Cox, Brian Howard, Josh De Cesare, Rich Williams, Dave Falkenburg, Daisie Huang, and Dave Radcliffe.

FIELD OF THE TECHNOLOGY

[0002] The field of technology relates generally to data processing systems, and more particularly but not exclusively to the management of power usage and temperature in the data processing systems.

BACKGROUND

[0003] A data processing system (e.g., a desktop computer or a laptop computer) typically contains a number of components that consume power from a power supply (e.g., battery or AC adapter) to perform different tasks. For example, a microprocessor consumes power to perform computation, generating heat in the process; and, a cooling fan consumes power to dissipate heat.

[0004] Typically, a data processing system is designed for operating in a given environment to deliver high computation performance. One or more fans and heat sinks are typically used to cool the system so that the data processing system is not overheated in a condition of normal use.

[0005] To be energy efficient, some computers have power management systems which may temporarily put a hard drive or a display screen in a low power mode after idling for a period of time. When a component is in a low power mode, the component is not functioning at least in part (e.g., the display screen is not displaying images, a hard drive cannot be accessed for read or write operations, and a section of a chip is not energized with power to perform

computation). In some systems, a cooling fan is triggered by a temperature sensor such that the cooling fan is turned on when the sensor detects that the temperature is above a threshold.

[0006] To protect from overheating, some microprocessors have built-in hardware to slow a processor when the processor is too hot. However, built-in hardware in a processor that slows down the processor when the processor is too hot is restricted to only changing processor performance to regulate the temperature. Intrinsically, it is not able to regulate other devices in the system or optimize thermal management of the entire system. Similarly, some computers (e.g., iBook laptops from Apple Computer, Inc.) automatically enter into a shut down when it is too hot (e.g., because a fan failed). Automatic shutdown of a notebook computer is an emergency solution for unusual situations, such as when the cooling fan is failing. It does not regulate the temperature during the normal use of the computer.

[0007] Thus, a computing platform (including a processor) is commonly designed for increased performance, which typically requires increased power consumption. However, computing platforms, especially in mobile applications, are also designed to reduce power consumption such that a limited power resource (e.g., a battery) can support the computing platform for an increased period of usage time. These design goals are typically in conflict.

[0008] One conventional solution to the conflicting design goals is to provide a means for a user to switch the configuration of the computing platform between a high performance mode and a power conservation mode, as desired. For example, a computing platform may allow a user to select the desired mode via a hardware switch or via a menu and dialog box displayed by the computing platform. For example, some computers allow a user to manually select a clock frequency for the microprocessor.

SUMMARY OF THE DESCRIPTION

[0009] Methods and apparatuses to manage working states of a data

processing system are described here. Some of the embodiments of the present invention are summarized in this section.

[0010] At least one embodiment of the present invention includes a data processing system with one or more sensors (e.g., physical sensors such as tachometer and thermistors, and logical sensors such as CPU load) for fine grain control of one or more components (e.g., processor, fan, hard drive, optical drive) of the system for working conditions that balance various goals (e.g., user preferences, performance, power consumption, thermal constraints, acoustic noise). In one example, the clock frequency and the core voltage for a processor are actively managed to balance performance and power consumption (heat generation) without a significant latency. In one example, the speed of a cooling fan is actively managed to balance cooling effort and noise (and/or power consumption).

[0011] Thermal managers according to embodiments of the present invention monitor the system temperature based on a number of sensors and conditions (e.g., sensed temperatures, lid position, battery charging status, current computation tasks and user preferences) to provide the best of mixture of cooling (e.g., by controlling one or more cooling fans) and reduced heat generation (e.g., by adjusting the working states of the heat generating devices, such as CPU, GPU, hard drives, optical drives, memory chips, core logic chips and others) to provide the best performance for the current task.

[0012] In one aspect of the present invention, a method to operate a data processing system includes: determining a control level for a first component of the data processing system based on information obtained from a plurality of sensors (e.g., a temperature sensor determining a temperature in the data processing system, such as a particular component's local temperature which is one of many components in the system); and, automatically adjusting the control of the first component according to the control level to move the first component from a first working state to a second working state. In one example, the first

component includes a cooling fan of the data processing system; and, the cooling fan runs at a first speed in the first working state and a second speed in the second working state; and, in one example, a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed. In one example, the first component includes a processor; the first working state includes a first clock frequency and a first core voltage for the processor; and, the second working state includes a second clock frequency (which may be lower or higher than the first clock frequency) and a second core voltage (which may be lower or higher than the first core voltage) for the processor. In one example, the first component includes a Graphics Processing Unit (GPU); the first working state includes a first swap interval; and, the second working state includes a second swap interval. In one example, the control of a second component is further adjusted automatically based on the information obtained from the plurality of sensors to move the second component from a third working state to a fourth working state. In one example, the first component is a heat source of the data processing system and the second component is a cooling source of the data processing system. In one example, the control level is determined further based on one or more user preferences. In one example, one of the sensors includes a software module (e.g., an operating system's kernel) determining a processor load of the data processing system.

[0013] In one aspect of the present invention, a method to operate a data processing system includes: determining a subset of control settings from a plurality of control settings of a plurality of components of the data processing system based on information obtained from a plurality of sensors (e.g., a temperature sensor, a tachometer, a software module determining a load of a processor), each of which determines an aspect of a working condition of the data processing system; and adjusting the subset of control settings to change working states of corresponding components of the data processing system to balance requirements in performance and in at least one of: thermal constraint and power

consumption. In one example, the plurality of components include heat sources (e.g., a Central Processing Unit (CPU), a Graphics Processing Unit (GPU), a hard drive, an optical drive, an Integrated Circuit (IC) bridge chip) of the data processing system and cooling sources of the data processing system. In one example, an amount of cooling change is determined based on the information obtained from the plurality of sensors; and, the subset of control settings are adjusted to effect the amount of cooling change. In one example, the amount of cooling change is determined according to a fuzzy logic; and, determining the subset of control settings includes determining a prioritized list of the plurality of control settings. In one example, the prioritized list is determined at least partially based on one or more user preferences. In one example, the amount of cooling change is parceled out to the subset of control settings. In one example, a first state of the data processing system is determined from the information obtained from the plurality of sensors; and, the subset of control settings is determined from a decision to move the data processing system from the first state to a second state.

[0014] In one aspect of the present invention, a method to operate a cooling fan of a data processing system includes: adjusting the cooling fan from running at a first speed to running at a second speed in response to a temperature sensor measurement and a user preference. In one example, it is further verified that the cooling fan is running at the second speed (e.g., using tachometer information obtained from a fan controller for the cooling fan). In one example, a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed. In one example, one or more temperature measurements are determined; and the second speed for the cooling fan is determined based at least partially on the one or more temperature measurements. In one example, the one or more temperature measurements are obtained from one or more temperature sensors instrumented in the data processing system; and, the one or more temperature measurements indicate temperatures of at least one of: a) a

microprocessor of the data processing system; b) a graphics chip of the data processing system; and c) a memory chip of the data processing system. In one example, the microprocessor of the data processing system determines the second speed. In one example, the second speed is determined further based on at least one of: a user preference stored in a machine readable medium of the data processing system; and, a computation load level on the data processing system (e.g., the load level is low because the processor is idling and the temperature level is low and a user preference has been set by a user such that in this state the fan's speed is reduced to reduce noise and power consumption).

[0015] In one aspect of the present invention, a method to operate a processor of a data processing system includes: shifting a power supply of the processor from a first voltage to a second voltage without resetting the processor. In one example, a frequency of a clock of the data processing system is slewed (changed slowly) to transit a clock of the processor from a first frequency to a second frequency (e.g., by instructing a clock chip to use a new frequency multiplier). In one example, the processor continues to execute instructions while the frequency of the clock is slewed and while the power supply is shifted from the first voltage to the second voltage. In one example, the power supply is maintained at one of the first and second voltages while the frequency of the clock is slewed; and, the clock of the processor is maintained at one of the first and second frequencies while the power supply is shifted from the first voltage to the second voltage. In one example, the first frequency is higher than the second frequency; the first voltage is higher than the second voltage; and, the power supply is shifted from the first voltage to the second voltage after the clock of the processor transits from the first frequency to the second frequency. In another example, the first frequency is lower than the second frequency; the first voltage is lower than the second voltage; and, the power supply is shifted from the first voltage to the second voltage before the clock of the processor transits from the first frequency to the second frequency. In one example, a frequency multiplier of the processor

is adjusted to switch a clock of the processor from a first frequency to a second frequency. In one example, the processor is not reset during switching from the first frequency to the second frequency.

[0016] The present invention includes apparatuses which perform these methods, including data processing systems which perform these methods, and computer readable media which when executed on data processing systems cause the systems to perform these methods.

[0017] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

[0019] **Figure 1** shows a block diagram example of a data processing system which may be used with the present invention.

[0020] **Figure 2** shows a data processing system according to one embodiment of the present invention.

[0021] **Figure 3** is a simplified block diagram illustrating heat sources and cooling sources of an exemplary data processing system having power and temperature management according to one embodiment of the present invention.

[0022] **Figure 4** is a simplified block diagram illustrating a data processing system depicted partitioned into thermal zones for power and temperature management according to one embodiment of the present invention.

[0023] **Figure 5** illustrates operational states for system level power management according to one embodiment of the present invention.

[0024] **Figure 6** illustrates operational states for processor and/or system power management according to one embodiment of the present invention.

[0025] **Figure 7** illustrates transitions from one run state to another run state according to one embodiment of the present invention.

[0026] **Figure 8** illustrates a detailed block diagram representation of a data processing system with active power and temperature management according to one embodiment of the present invention.

[0027] **Figure 9** shows a software module diagram which shows software to manage the operation state of a data processing system according to one embodiment of the present invention.

[0028] **Figure 10** illustrates an example of a method to determine actions to be performed using fuzzy logic in operating a data processing system according to one embodiment of the present invention.

[0029] **Figures 11** and **12** illustrate an example defuzzification method to merge different actions as one quantified action to operate a data processing system according to one embodiment of the present invention.

[0030] **Figure 13** shows a software module diagram which shows software to manage the operation state of a data processing system according to one embodiment of the present invention.

[0031] **Figures 14 – 16** show methods to operate a data processing system according to embodiments of the present invention.

[0032] **Figure 17** illustrates a method to parcel out cooling changes to a number of controls according to one embodiment of the present invention.

[0033] **Figure 18** illustrates an example of a state diagram which shows a way to operate a data processing system according to one embodiment of the present invention.

DETAILED DESCRIPTION

[0034] The following description and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to provide a thorough understanding of the present invention. However, in certain instances, well known or conventional details are not described in order to avoid obscuring the description of the present invention. References to one or an embodiment in the present disclosure are not necessarily

references to the same embodiment; and, such references mean at least one.

[0035] As the performance of data processing systems continues to increase, so do power and cooling requirements. A fan and heat sink may not be adequate for such high performance data processing systems. To meet the challenge of combined conflicting design goals (e.g., computation performance, power usage, cooling, acoustic noise, and others), at least one embodiment of the present invention seeks to manage the working states of various components of a data processing system according to sensed information (e.g., one or more temperature sensors and performance sensors and fan speed sensors). Since hardware-based solutions have pre-determined flexibility, at least one embodiment of the present invention utilizes the processing power of the data processing system to provide software-based management solutions (e.g., software in a kernel of an operating system).

[0036] In the present document, a working state of a component (e.g., a microprocessor or a fan) is a state in which the component works to provides the functionality of the component at a specific level of cost (e.g., the consumption of power usage, the generation of noise or heat, or other factors). A working state does not normally include the state in which the component does not work to provide its functionality for the system.

[0037] One embodiment of the present invention involves power and thermal management strategies to meet the combined challenge of high performance, low power consumption, low noise and tight thermal constraints. In one embodiment, power and temperature in a computer are actively managed so that the computer can go faster, run quieter with extended battery life, and avoid running too hot, as the computation speeds increase and the enclosures of computers continue to push the limits of engineering.

[0038] In one embodiment of the present invention, a computer system is instrumented with one or more sensors; and, at least one component of the system has a number of different working states. For example, different working states

have different power consumption levels and performance levels (e.g., processor speeds measured in megahertz or processing operations per second, etc.), which are actively managed to meet conflicting goals, such as high performance and low power consumption, subject to thermal constraints (e.g., the interior of the computer enclosure should not or cannot exceed a certain temperature which may damage certain components in the enclosure). Managing thermal output and finessing cooling efforts can help some machines avoid the need for fans, while allowing other machines to run fans more quietly. With the help of the information collected from the sensors, the computation performance, user preferences and environmental requirements can be balanced to reach a best mix for a particular usage of the system.

[0039] In one embodiment of the present invention, sensors are instrumented (e.g., in the hot spots for measuring temperature); controls are constructed to gracefully adjust the working states (e.g., through the adjustment of frequencies and voltages) for tradeoff in performance, power consumption, heat generation and heat removal; and, a thermal manager is provided to monitor and control one or more thermal zones, according to the constraints of system and user preferences, which define the priorities of conflicting goals.

[0040] In one embodiment of the present invention, temperature sensors are instrumented so that the temperatures of hot spots can be periodically polled. The temperature sensors can be implemented as thermal diodes on Integrated Circuit (IC) chips (e.g., microprocessors, graphics chips, microcontrollers, and others). Further, tachometers are instrumented to obtain the feedback about the working states of fans.

[0041] In one embodiment of the present invention, fine-grained control of frequencies and voltages for a data processing system is added in an architecture-independent way to manage power consumption and heat generation. For example, Central Processing Unit (CPU) voltage and frequency control are provided to allow multiple CPU frequency and voltage states with fine-grained

control beyond just high or low; and cooling fans have speed control beyond just on or off.

[0042] In one embodiment of the present invention, software device drivers dynamically tweak power and performance. For example, a CPU software driver manages CPU working states (e.g., speed, frequency, voltage) based on computation load, sensor measurements (e.g., CPU temperatures and CPU load levels), and various preferences and priorities (e.g., user preferences with respect to fan noise or other noise or battery life). Device drivers for other controls use a similar approach to select the working state based on the required work load and various constraints.

[0043] In one embodiment of the present invention, a thermal manager software module controls the power consumption level of various components through the software device drivers. For example, the software thermal manager may monitor and control physical (e.g. temperature) and logical (e.g. CPU load) sensors, optimize for user-center or design-center priorities, such as performance, heat, battery life, and noise, force drivers into lower power states to minimize power consumption and/or heat production, and remove heat with minimal noise. Further, the thermal manager may monitor and control multiple independent zones.

[0044] It is vastly cheaper to reject a faulty part during a factory burn in process (before a customer receives the part) than to handle a customer return. After a design is instrumented, bad parts can be detected early during the manufacture of or testing of the system, using diagnostics tools. For example, when a computer is instrumented with temperature sensors, misapplied heatsinks may be detected for correction, removing one of the most costly manufacturing defects.

[0045] Many of the methods of the present invention may be performed with a digital processing system, such as a conventional, general-purpose computer system. Special purpose computers, which are designed or programmed to

perform only one function, may also be used.

[0046] **Figure 1** shows one example of a typical computer system which may be used with the present invention. Note that while **Figure 1** illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will also be appreciated that network computers and other data processing systems which have fewer components or perhaps more components may also be used with the present invention. The computer system of **Figure 1** may, for example, be a Macintosh computer from Apple Computer, Inc.

[0047] As shown in **Figure 1**, the computer system 101, which is a form of a data processing system, includes a bus 102 which is coupled to a microprocessor 103 and a ROM 107 and volatile RAM 105 and a non-volatile memory 106. The microprocessor 103, which may be, for example, a G3 or G4 microprocessor from Motorola, Inc. or IBM or a G5 microprocessor from IBM is coupled to cache memory 104 as shown in the example of **Figure 1**. The bus 102 interconnects these various components together and also interconnects these components 103, 107, 105, and 106 to a display controller and display device 108 and to peripheral devices such as input/output (I/O) devices which may be mice, keyboards, modems, network interfaces, printers, scanners, video cameras and other devices which are well known in the art. Typically, the input/output devices 110 are coupled to the system through input/output controllers 109. The volatile RAM 105 is typically implemented as dynamic RAM (DRAM) which requires power continually in order to refresh or maintain the data in the memory. The non-volatile memory 106 is typically a magnetic hard drive or a magnetic optical drive or an optical drive or a DVD RAM or other type of memory systems which maintain data even after power is removed from the system. Typically, the non-volatile memory will also be a random access memory although this is not required. While **Figure 1** shows that the non-volatile memory is a local device

coupled directly to the rest of the components in the data processing system, it will be appreciated that the present invention may utilize a non-volatile memory which is remote from the system, such as a network storage device which is coupled to the data processing system through a network interface such as a modem or Ethernet interface. The bus 102 may include one or more buses connected to each other through various bridges, controllers and/or adapters as is well known in the art. In one embodiment the I/O controller 109 includes a USB (Universal Serial Bus) adapter for controlling USB peripherals, and/or an IEEE-1394 bus adapter for controlling IEEE-1394 peripherals.

[0048] Sensors 112 are coupled to controller 109 to provide information about the operating environment condition of the components of the data processing system. For example, sensors 112 may include temperature sensors for determining the temperatures at a plurality of locations in the data processing system, such as the temperatures of microprocessor 103, volatile RAM 104, a hard drive and an optical (e.g., CD/DVD) drive; sensors 112 may further include a fan tachometer for determining the speed of a cooling fan, a light sensor for determining the amount of required backlight; a sensor to determine whether a display of a laptop is opened or closed and others. Although **Figure 1** illustrates a configuration in which sensors 112 are coupled to controller 109, it is understood that sensors may also be integrated into components (e.g., microprocessor 103). Further, software sensors like kernel load factor are also used in at least some embodiments of the present invention.

[0049] It will be apparent from this description that aspects of the present invention may be embodied, at least in part, in software. That is, the techniques may be carried out in a computer system or other data processing system in response to its processor, such as a microprocessor, executing sequences of instructions contained in a memory, such as ROM 107, volatile RAM 105, non-volatile memory 106, cache 104 or a remote storage device or a combination of memory devices. In various embodiments, hardwired circuitry may be used in

combination with software instructions to implement the present invention. Thus, the techniques are not limited to any specific combination of hardware circuitry and software nor to any particular source for the instructions executed by the data processing system. In addition, throughout this description, various functions and operations are described as being performed by or caused by software code to simplify description. However, those skilled in the art will recognize what is meant by such expressions is that the functions result from execution of the code by a processor, such as the microprocessor 103.

[0050] A machine readable medium can be used to store software and data which when executed by a data processing system causes the system to perform various methods of the present invention. This executable software and data may be stored in various places including for example ROM 107, volatile RAM 105, non-volatile memory 106 and/or cache 104 as shown in **Figure 1**. Portions of this software and/or data may be stored in any one or more of these storage devices.

[0051] Thus, a machine readable medium includes any mechanism that provides (i.e., stores and/or transmits) information in a form accessible by a machine (e.g., a computer, network device, personal digital assistant, manufacturing tool, any device with a set of one or more processors, etc.). For example, a machine readable medium includes recordable/non-recordable media (e.g., read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; etc.), as well as electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0052] **Figure 2** shows a data processing system according to one embodiment of the present invention. In **Figure 2**, the data processing system includes one or more processors 201 (e.g., a Graphics Processing Unit (GPU) and one or more Central Processing Units (CPU)), devices 221 – 229 (e.g., cooling fan, battery charging system, AC adapter) and machine readable media 209 (e.g., RAM chips, ROM chips, hard drive, optical drive). The data processing system

may include communication links to one or more devices outside housing 200. For example, the data processing system may be connected to a network or peripheral devices (221), such as a monitor, a keyboard, a cursor controlling device (e.g., mouse, a track ball, or a touch screen, or a touch pad), a printer, a storage device, or others. Although **Figure 2** shows that machine readable media 209 are inside housing 200, it is understood that a portion of the machine readable media may be outside housing 200 (e.g., connected through an IEEE 1394 bus or a USB bus, or a network connection). Further, some of the peripheral devices listed above as examples of devices 221 may be integrated inside housing 200. For example, a touch pad, an LCD display panel and a keyboard can be integrated within housing 200 (e.g., on a notebook computer).

[0053] Machine readable media 209 have program instructions and data for operating system 233 (e.g., Mac OS X), application programs 231 (e.g., a word processing program), and operating manager 235 for managing the states of the components of the system. Note that operating manager 235 can be a part of operating system 233. Sensors 211 – 219 are instrumented within housing 200 of the data processing system to obtain information about the environmental conditions of various components of the data processing system, such as the temperature of the processors (201), the fan speed of a cooling fan, the lid position (e.g., open or closed). Some of the sensors can also be software modules that determine the computation loads for the processors. Operating manager 235, when executed on at least one of the processors (201), causes the processing of the information obtained from sensors 211 – 219 and the adjustment of the working states of the processors (201) and the devices (e.g., 221 – 229) to provide trade-off between performance and power usage while maintain proper thermal constraints to avoid damage or loss of data.

[0054] Because software can crash, a safety thermal cutoff remains independent of the operating system and is tied directly to Power Management Unit (PMU 205) which includes a hardware only portion which is not effected by

a software crash. When a sensor (e.g., 207) detects a temperature that exceeds a safety threshold, hardware-based safety trigger 203 signals PMU for proper action. PMU 205 may force a shutdown of the data processing system to prevent damage when an extreme thermal condition is detected. PMU 205 notifies processors 201 about the safety threshold; and, a software module may then notify the user that the safety threshold is being approached, if one of processors 201 is responsive to PMU. If none of processors 201 is responding to the safety alert or if its response is inadequate to reduce the temperature, PMU 205 takes actions to power off the data processing system. Note that sensor 207 may be different and separate from sensors 211 – 219; alternatively, sensor 207 may be part of sensors 211 – 219 as shared hardware.

[0055] **Figure 3** is a simplified block diagram illustrating heat sources and cooling sources of an exemplary data processing system having power and temperature management according to one embodiment of the present invention. In **Figure 3**, data processing system 300 has power and temperature management according to one embodiment of the present invention. System 300 includes power manager 301 (e.g., a software program running as a portion of the operating system on data processing system 300), heat sources 311 – 317 (e.g., Central Processing Unit (CPU), Graphics Processing Unit (GPU), memory chips, hard drive, optical drive, backlight, battery charging system, system core logic and others) and cooling sources 321 – 324 (e.g., cooling fans, heat pipes and heat sinks) located within housing 300 of the data processing system. The data processing system is instrumented with sensors 331 – 335 (e.g., temperature sensors, tachometers, light sensors, noise sensor and others) within housing 300.

[0056] In one embodiment of the present invention, power manager 301 is physically and/or logically coupled to heat sources 311 – 317, cooling sources 321 – 324 and sensors 331 – 335. For example, power manager 301 can be partially a software program running on the data processing system, using the processing power and storage capacities provided by the heat sources, and

partially hardware providing control signals to the heating sources and cooling sources to balance the performance and power consumption and limit the temperatures monitored by the sensors. Some of sensors 331 – 335 may be disposed proximate to some of heat sources 311 – 317, with some of cooling sources 321 – 324 being disposed so as to operatively coupled to the heat sources to remove heat from the heat sources.

[0057] **Figure 4** is a simplified block diagram illustrating a data processing system depicted partitioned into thermal zones for power and temperature management according to one embodiment of the present invention. In **Figure 4**, the data processing system is thermally divided into a plurality of zones (e.g., zones 401 – 404). The temperature of these thermal zones can be individually controlled via controls (e.g., CPU controller, fan controller, hard drive controller) to the heat sources and cooling sources disposed within each thermal zone using the information from the sensors in the corresponding zone.

[0058] In one embodiment of the present invention, the data processing system is enclosed within multiple isolated thermal zones (MITZ). In the present application, a thermal zone is a volume of space containing components that have strong thermal interaction. For example, in a thermal zone, the heat from one device may raise significantly the temperature of another; the devices may share a common cooling system or a common temperature sensor. One or more temperatures within one zone are sensed for thermal management in that zone.

[0059] It is possible that one zone in a system requires software management, and another does not. For example, a power supply could control a fan in a first zone, with the CPU and another fan in a second zone. In this example, cooling the power supply may very well cool the rest of the zone adequately (perhaps even over-cooling some components). Since power consumption relates to heat in the power supply, it might be possible that meeting the cooling requirements of the power supply is guaranteed to cool all other devices in its thermal zone. In this case, the power supply zone would require no active management. At the same

time, the CPU zone would need sensors and active thermal management.

[0060] **Figure 5** illustrates operational states for system level power management according to one embodiment of the present invention. There are four basic states in the system level power state diagram. These states are Off 507, Run 501, Idle 509 and Sleep 503. Run 501 includes a range of working states with different power and performance levels.

[0061] Off 507 represents a state when all power plane for the system is turned off, with the exception of those power planes necessary to run the power system (and one or more peripherals that can be operated in an autonomous mode).

[0062] Idle 509 represents a state when the system is idling. Cache coherency is maintained when in Idle 509. All clocks are running and the system can return to running code within a few nanoseconds. When all CPUs stop computing, the system enters (519) from Run 501 into Idle 509, such as when the last active processor of the system is stopped in the Nap state 603 and is dynamically switched between the Nap state 603 and the Doze state 607 for snoop cycles to complete. The North Bridge can control the switching between the Nap state 603 and the Doze state 607, which will be described below with **Figure 6** in detail. Any processor interrupt returns (519) the system from Idle 509 to Run 501.

[0063] Sleep 503 represents a state when the system is shutdown with various states preserved for instance recovery (513) to Run 501. In one embodiment, the states of the North Bridge, RAM and South Bridge are preserved to provide the appearance of instant-on/always-available. All processors are powered off after their caches are flushed and their states preserved in RAM for the system to enter (513) from Run 501 into Sleep 503. Other devices, such as the devices in PCI slots (or PCMCIA slots) are also powered off after preserving their state in RAM. All clocks in the system in the system are stopped except for the one for real-time clock function.

[0064] Run 501 represents a state when a least one processor of the system is

running. If the system has multiple processors, the individual processors may move in and out of their respective processor power states (e.g., Doze 607, Nap 603, Sleep 605, Off 609 in **Figure 6**) in a fully independent manner. While in Run 501 state, the processors can move together between different performance levels. The operating system sets the policies for when to shift performance levels and when to power manage individual CPUs. For example, if there are threads waiting to be scheduled, the system is in Run 501 state and at least one CPU is running and executing the threads. When the scheduler of the operating system runs out of threads, the system transitions from Run 501 to Idle 509.

[0065] There are multiple power and performance levels associated with Run 501, in which the system moves (511) among them as load and other constraints dictate. In one embodiment of the present invention, the processor continues executing instructions while shifting in performance levels, remaining in Run 501 but shifting between different working states with different power consumption levels and performance levels. For example, when more performance is needed, the CPU voltage and frequency may be increased to trade increased power consumption for high performance. When full performance is not needed and/or cooling down is required, the CPU voltage and frequency may be decreased to trade performance for reduction in power consumption, which helps cooling the processor(s) down.

[0066] In one embodiment of the present invention, a device is placed into a working state that uses as only much power as is required to perform a task. The power consumption for a device is scaled to match the work being requested. For example, the speed of a fan is adjusted to keep the temperature within the allowable range while using less power and producing less noise than when the fan is at the full speed. The fan is controlled to run only as fast as it needs to, and no faster. Sensors instrumented in the system are used to gather the information; and, the data processing system processes the information to decide how to control the devices.

[0067] In a given system implementation, there are different devices (e.g., CPU, hard drive, optical drive, graphics chip, power supply, memory chip, core logic) that may reach their thermal maximums (maximum thermal thresholds). If a device may be above its allowable temperature (to become overheated) under certain operating condition, the device may be included in a thermal management algorithm; and, a temperature sensor can be instrumented for detecting the current temperature of the device. If the enclosure and airflow somehow guarantees that a device will stay within its specifications (including noise), temperature sensing of that device is not required; and, that device may be included only for power management.

[0068] There also might be a case where the temperature of one device is sensed indirectly via its effect on another device. For example, a design might have the hard drives preheating the air that flows over the CPU. If the algorithm for protecting the CPU always turns on the fan before the hard drives overheat, it is not necessary to sense hard drive temperature directly.

[0069] In one embodiment of the present invention, device management includes using sensors for the devices and implementing software drivers that controls the operations of the devices according to the work being requested. In one embodiment, sensors include thermistors, hard drive temperatures, kernel loads, and more. Controls are provided to vary fan speeds, backlight brightness, hard drive speeds, to power devices off, to put a CPU in a different working state, and to throttle thread scheduling.

[0070] Many of the devices of a computer (e.g., CPUs, GPU, hard drives, optical drives, backlight, charging system, etc.) have multiple power states, which can be used for active power management, such as varying spindle speed of hard drives.

[0071] **Figure 6** illustrates operational states for processor and/or system power management according to one embodiment of the present invention. In the Run state 601, all processor units are active. Dynamic clock stopping to

individual functional units is allowed, but are invisible to software. Clocks are automatically restored to functional units immediately upon detection of any instruction to be dispatched for that functional unit.

[0072] In the Doze state 607, all execution units are stopped. All processor caches and bus interface logic necessary to maintain cache coherency are active.

[0073] In the Nap state 603, all execution units are stopped, along with all caches and bus interface logic. The system can bring (621) the processor back to the Doze state 607 by de-asserting a signal (e.g., QAck). This is done when the system needs the processor to perform cache coherency operations.

[0074] In the Sleep state 605, all execution units are stopped, while the processor state is maintained. Before entering (619) the Sleep state 605, all processor caches are flushed, as a sleeping processor does not perform any cache coherence operation. In systems with multi-drop bus topologies (60x, MaxBus), sleeping processors do not respond to deassertion (e.g., QAck). Systems with point-to-point interconnect topologies (e.g., ApplePI) uses per-processor deassertion signals, but not to a sleeping processor.

[0075] In the Off state 609, core power is removed from the processor. The Off state 609 is typically used in multiprocessor systems. In one embodiment, before a processor enters the Power Off state, it sets its PowerDownEnabled bit inside the core logic. Once the core logic asserts deassertion signal (e.g., QAck) to put a processor into the Sleep state 605, the core logic checks the PowerDownEnabled bit. If the bit is set, the processor interface is then put into an appropriate condition for powering down the processor. An interrupt is generated to signal the system that the processor is ready to have its power removed. A live processor receives the interrupt and performs the appropriate actions to remove power from the sleeping CPU, moving (615) it into the Off state 609.

[0076] When software determines that it again needs the processor that is in the Off state 609, it performs the appropriate actions to reapply power to the CPU and reset it, returning (613) it to the Run state 601.

[0077] In a single processor system, the CPU may avoid the Sleep state 605 and target the Nap state 607 to avoid the penalty associated with cache flushing. In a multi-processor system, every processor except the last active processor uses either the Doze state 607 or the Sleep state 605. These two states impose no penalty for performing snoop operations. The last active processor will use Nap 607 in a manner similar to the single processor. The system enters the Idle state 509 when the last active (or the only) processor enters the Nap state 603.

[0078] One embodiment of the present invention moves the processor from performance level to performance level by changing frequency and power voltage while staying within the Run state 601. Each performance level is a different combination of CPU core frequency, CPU bus frequency, and CPU core voltage that defines both a computing performance level and a power consumption level. Each performance level representing a working state of the CPU.

[0079] The number of performance levels implemented in a system depends upon a variety of factors. For example, some CPUs may only support two operating points, limiting the system implementation to Run 0 and Run 1. A well-managed portable design may implement Run 0 for minimum power, Run 1 for specific functionality level such as DVD playback, and Run 2 for the clock frequency for the portable system. A high performance desktop may implement Run 0 for power and thermal savings, Run 1 at 90% of maximum clock frequency for most operations, and Run 2 at maximum power. One example of different performance levels is illustrated below.

[0080] Run 0: This is the lowest power and performance level supported by the system. The processor's Doze mode entered from Run 0 may be different than that used at the other performance levels. In particular, a large portion of the CPU may be powered at a lower voltage than that required by the snoop logic, saving significant leakage power. Run 0 can be the state at which the system starts executing code after a Power-On or Restart event.

[0081] Run 1 ... n: These states have incremental power and performance

levels above the next lower level one. These states differ solely by CPU core voltage and frequency, which sets leakage power, operating power and CPU performance.

[0082] Run n+1: This state is special in requiring no transitions out of this state. Some systems may achieve higher performance if the CPU does not leave the Run state 601. Since the surge currents for transition in and out of Run can be significant, transitions can cause the maximum droop on the power supply rail. By avoiding these droops at the highest performance point, the maximum performance can be achieved. However, by not allowing the CPU to Nap (603) or Sleep (605), significant additional power will be used.

[0083] In one embodiment, with the processor environment set to a specific performance level, the processors can independently transition back and forth between their different power saving states (Run 601, Doze 607, Nap 603, Sleep 605, and Off 609). The system enters automatically from Run 501 into Idle 509 whenever there are no processors in the Run state 601.

[0084] Some operating systems (e.g., Mac OS X) have the capabilities for monitoring the activity level of the CPU(s). In one embodiment, after determining the processing load currently required by the CPU, the CPU can determine if it is above a certain activity level. Based on the processing load level, the management system can initiate a shift upward or downward in performance.

[0085] Likewise, if the CPU determines that it doesn't need all the performance that is currently available, it can shift downward in performance in order to save significant power. Since the CPU continues to operate throughout the process of shifting, no significant latency is incurred as a result of the performance level change.

[0086] In one embodiment, the algorithms used to determine when to shift up and down in performance are contained in system software. The software makes the decisions about when to change the performance level. In one embodiment of the present invention, the software relies not only on CPU load, but also on

system thermal conditions, user preferences and others, to decide whether to move to a higher or lower performance level.

[0087] In one embodiment of the present invention, the CPU continues executing instructions throughout the performance transition process without CPU reset.

[0088] **Figure 7** illustrates transitions from one run state to another run state according to one embodiment of the present invention. In one embodiment of the present invention, the transitions between the different CPU run levels are accomplished using a combination of frequency and voltage control. To increase performance (e.g., to move from Run 707 to Run 705), the CPU core voltage is first increased (701) to the level appropriate for Run 705. The CPU power supply moves the voltage between the two voltage points at a rate slow enough such that it does not induce errors in the running CPU. When the voltage transition is complete, the CPU is currently operating at point 709 in **Figure** 7; and, the CPU power supply may signal the system with an interrupt, indicating that it is now safe to increase frequency. At this point, the clock source starts to transition to the new (faster) operating frequency. The clock source slews (slowly changing) the frequency between the two operating points (e.g., 709 and 705) slow enough such that all the Phase Lock Loops (PLL) in the system can track the clock without causing errors in the running system.

[0089] Similarly, to decrease performance (e.g., to move from Run 705 to Run 707), the clock source is first instructed to move to the new (slower) operating frequency. The clock source again slews the frequency between the two operating points slow enough such that all the PLLs in the system can track the clock without causing errors in the running system.

[0090] Since the operation of slewing the frequency from point 705 to 709 takes a deterministic amount of time, it is possible to have the CPU wait the appropriate delay and then infer that it is operating at point 709. Alternatively, the clock chip can generate an interrupt after achieving the new frequency.

[0091] Once point 709 is reached, the CPU core voltage is decreased to the level appropriate for Run 707. The CPU power supply moves the voltage between the two voltage points at a rate slow enough such that it does not induce errors in the running CPU. Since no further action needs to be taken, the voltage transition may be left to complete unmonitored.

[0092] In a typical system, the different PLLs in the system have certain inter-relationships. For example, each PLL associated with the CPU may run at a frequency derived from a single, common reference clock. Each of these PLLs also has its own specified minimum and maximum operating frequencies. The individual PLL operating minimum and maximum frequencies imply that each PLL also has a specific minimum and maximum reference frequency that it will accept. In order for the system to work correctly, the reference clock must obey all the individual reference clock minima and maxima. In some cases, the PLL of a device (e.g., core logic) may be reprogrammed during the frequency transition operation to obey the clock minima and maxima of the device.

[0093] **Figure 8** illustrates a detailed block diagram representation of a data processing system with active power and temperature management according to one embodiment of the present invention. In **Figure 8**, the data processing system contains system core logic 801 (North Bridge), which interconnects CPU 805 and RAM 809. I/O controller 803 (South Bridge) connects core logic 801 with hard drive 811 and optical drive 813 (e.g., CD ROM, DVD ROM, CD R, CD RW, DVD R, or DVD RW) and other I/O devices (e.g., a keyboard, a cursor control device, or others, not shown in **Figure 8**). Some components (e.g., CPU 805, GPU 807 and RAM 809) may have elevated temperature after generating significant amount of heat during operation. Some components (e.g., hard drive 811 and optical drive 813) consume more power at a high speed and less power at a lower speed. Heat pipe 825 moves heat from one location to anther to transfer heat; heat sink 823 absorbs heat to regulate temperature; and, under the control of fan controller 845, variable speed fans 819 and 821 can work at different speeds

for tradeoff between the rate of cooling and the associated cost (e.g., noise and power consumption). Sensors 831 and 833 monitor the temperate of GPU 807 and CPU 805 for active management of the system according to embodiments of the present invention.

[0094] **Figure 8** shows a particular configuration for the illustration purpose. It is understood that different configurations can also be used with various methods of the present invention.

[0095] In one embodiment, CPU power 843 is controllable to move the core voltage of CPU 805 from one point to another for the shifting of power consumption level (e.g., along path 701 in **Figure** 7); and, clock source 841 is controllable to slew the core frequency of CPU 805 from one point to another for the shifting of performance level (e.g., along path 703 in **Figure 7**). When the CPU is working at a lower frequency, the CPU voltage is reduced to save power and reduce heat generation.

[0096] In **Figure 8**, sensor 831 measures the temperature of GPU 807; thus, GPU 807 and sensors 831 are in thermal zone 853. CPU 805, fan 819 and sensor 833 are thermally coupled in thermal zone 851. Fan 821 may cools power unit 817 and other components sufficiently such that other components may not need active thermal management.

[0097] Although the software-based thermal and power management can manage the operations of the system according to combined goals to achieve a best mix, software may crash. Thus, a hardware-based failsafe mechanism is used in one embodiment of the present invention as a backup. For example, sensor 835 may trigger a safety alert when a safety threshold for temperature is reached or exceeded. Note that sensor 835 may be replaced by an output from sensor 833 or 831. Alternatively, sensor 835 may be a circuit which combines the output of sensors 831 and 833 for safety trigger. In one embodiment of the present invention, the temperature sensors can trigger operations (e.g., force the fan to run at the full speed or a shutdown of the system) to prevent thermal runaway

(overheating to cause damage), when the temperature is above a threshold. In one embodiment, the sensor sends a signal to the power management unit (PMU 815) when this happens; and, PMU 815 controls power unit 817 for emergency powering off to prevent permanent damage.

[0098] In one embodiment of the present invention, software (e.g., a part of the operating system) has the responsibility for keeping all components within their respective thermal specification. The hardware failsafe is intended to prevent a crashed system from destroying itself. As such, the failsafe threshold may be set above a device's maximum operating temperature, as long as it is still below the threshold above which permanent damage may occur.

[0099] In one embodiment, PMU 815 implements a forced system shutdown function. When a hardware failsafe trips (e.g., by sensor 835), PMU 815 tries to determine if the CPU is active. One method of determining if the CPU is dead is to use a watchdog timer (e.g., in KeyLargo/K2) to determine whether the CPU response to a signal within a specified time period set for the timer. If the CPU is dead (e.g., not responding to the signal before the timer expires) after the failsafe has tripped, PMU 815 shuts off power to the system.

[00100] In one embodiment, when the failsafe triggers, the information about the failsafe shutdown is recorded so that a user can find out about the event at the next boot of the system. If a machine refuses to boot because it detects a misapplied heatsink, the LED emits a code to convey the nature of the failure. Further, when the system cannot be sufficiently cooled, and devices are set to slower operating speeds, the user might be informed that the system's performance is suffering because of the heat.

[00101] Heatsinks (e.g., 823) generally provides a valuable time to respond to non-responsive systems. Conversely, misapplied heatsinks can lead to destruction of a CPU in mere seconds. In one embodiment, during boot time, PMU 815 has a thermal trip watchdog timer that shuts off a non-responsive system with a time period before an unsinked device melts down.

[00102] In one embodiment of the present invention, the boot ROM code sets the thermal trip points in the thermal sensors for devices that might sustain permanent damage if over a critical temperature for more than a few seconds (e.g. CPUs).

[00103] In one embodiment of the present invention, software manages a thermal zone (e.g., 851) by controlling one or more fans (e.g., fan 819 or other "cooling" devices, such as CPU 805 for reduced heat generation) in that zone. The fan control is in the form of speed control, and not simple on or off operations. For example, a signal is sent to control the fan speed by varying duty cycle.

[00104] In addition to telling the fan how fast to spin, the software detects whether or not the fan has responded to reliably manage the thermal zone. For example, the software can used the tachometer input from fan controllers to obtain this feedback.

[00105] In one embodiment of the present invention, the tachometer feedback is to ensure that the fan begins spinning when first turned on. Fans may run at speeds slower than those required for spin-up. Some fan controllers start the fan at full speed and then backs off. In one embodiment, a better algorithm for spinning up a fan is to slowly increase the duty cycle until the fan starts up, and then slowly back it down to the actual desired fan speed (if slower than the startup speed). Tachometer feedback can be used to implement this algorithm.

[00106] In one embodiment of the present invention, a fan is controlled by a fan curve, which describes the relationship between the temperature values of one or more temperature sensors and the speed of the fan. For example, one fan curve specifies that for a given temperature the fan is to run a given speed. In one embodiment, there is one fan curve to map each senor to each fan for a given performance level; thus, the total number of fan curves is:

[00107] (number of sensor and performance level combinations) × (number of fans)

[00108] The manager select the maximum speed from the speeds required by all sensors according to the fan curves at the current performance level as the desired fan speed.

[00109] In one embodiment of the present invention, the fan curves are determined based on temperature measurements taken at different fan speeds and at different performance levels.

[00110] In one embodiment of the present invention, software controls fan speed to manage system thermals. The goal of the software is to run the fan, as slowly and quietly as possible while maintaining device specification. Since a single thermal zone may have multiple hot spots, the software runs the fan at the slowest speed required to keep all hot spots in check, even if it means that one device is cooled more than required in order to keep another device within the specification.

[00111] In one embodiment of the present invention, Active Thermal Management software is implemented in the kernel of an operating system. A thermal manager processes input data, including sensed information (e.g., temperature, CPU processing load, GPU processing load), detected conditions (e.g., battery charging, lid closed, sleep mode) and user preferences (e.g., prefer high graphics processing, prefer low noise) to optimize and direct accordingly CPU and/or GPU processing levels, battery charging periods, fan speeds and drive performance. Thus, the management system integrates the inputs from sensors, user preferences, current tasks and other conditions like lid closed operation to determine the optimum way to keep the temperature of the system within a desired range by increasing the cooling or decreasing the heat produced. In at least one embodiment, users of the data processing system can manage their own thermal solutions through the thermal management software modules implemented on the data processing system (e.g., by setting the user preferences).

[00112] **Figure 9** shows a software module diagram which shows software to manage the operation state of a data processing system according to one

embodiment of the present invention. In one embodiment of the present invention, a thermal management software system includes several modules, including sensor driver 907, thermal manager 901, and control driver 903. Thermal manager 901 performs the central decision-making. Sensor driver 907 communicates with device driver 909 to obtain sensed information; and, control driver 903 communicates with device driver 905 to adjust working states of one or more components (e.g., CPU, GPU or fan). In one embodiment of the present invention, device information 911 from the device tree 913 and Boot ROM 915 are collected for the instantiation of sensor driver 907 and control driver 903 during the initialization period.

[00113] In one embodiment of the present invention, the thermal manager monitors and controls the internal temperatures of the data processing system, on which the operating system is running, to prevent uncomfortable or unsafe temperatures. Certain parts, like the processor and graphics hardware, are more prone to overheating than others. Other components, like optical and hard drives, may fail due to excessive heat in the system. In order to monitor the temperatures of particularly hot components, the thermal manager (901) obtains temperature information about them from sensor drivers (e.g., 907). Based on the temperature information, the operating system instructs control drivers (e.g., 903) to take action to mitigate temperature increases as necessary in a coherent fashion.

[00114] In one embodiment of the present invention, thermal manager 901 contains a set of global rules that dictate how to manage the system: whether to manage more heavily, stay the same, or manage less heavily. For example, a thermal manager may contain a Cooling Decider to determine the amount of cooling adjustment required based on the information obtained from sensor driver 907, a priority decider to prioritize a list of controls according to user preferences 921, system information 923, and a control decider to adjust the controls according to the prioritized list of controls to achieve the determined amount of cooling adjustment. For example, the Cooling Decider takes sensor value data

of the present invention. In one embodiment of the present invention, a thermal management software system includes several modules, including sensor driver 907, thermal manager 901, and control driver 903. Thermal manager 901 performs the central decision-making. Sensor driver 907 communicates with device driver 909 to obtain sensed information; and, control driver 903 communicates with device driver 905 to adjust working states of one or more components (e.g., CPU, GPU or fan). In one embodiment of the present invention, device information 911 from the device tree 913 and Boot ROM 915 are collected for the instantiation of sensor driver 907 and control driver 903 during the initialization period.

[00113] In one embodiment of the present invention, the thermal manager monitors and controls the internal temperatures of the data processing system, on which the operating system is running, to prevent uncomfortable or unsafe temperatures. Certain parts, like the processor and graphics hardware, are more prone to overheating than others. Other components, like optical and hard drives, may fail due to excessive heat in the system. In order to monitor the temperatures of particularly hot components, the thermal manager (901) obtains temperature information about them from sensor drivers (e.g., 907). Based on the temperature information, the operating system instructs control drivers (e.g., 903) to take action to mitigate temperature increases as necessary in a coherent fashion.

[00114] In one embodiment of the present invention, thermal manager 901 contains a set of global rules that dictate how to manage the system: whether to manage more heavily, stay the same, or manage less heavily. For example, a thermal manager may contain a Cooling Decider to determine the amount of cooling adjustment required based on the information obtained from sensor driver 907, a priority decider to prioritize a list of controls according to user preferences 921, system information 923, and a control decider to adjust the controls according to the prioritized list of controls to achieve the determined amount of cooling adjustment.

For example, the Cooling Decider takes sensor value data and calculates how much it should turn the system cooling up or down. Specific rules that represent additional criteria, such as user preferences and environmental factors, are used to prioritize the available controls for use in the determination of control indices. The Priority Decider creates a sorted list of controls ranked in the order they should be changed according to these rules. The Cooling Decider passes the desirable cooling change and any relevant information, such as the target thermal zone in which the cooling change is to be implemented, to a Control Decider, which implements the desirable cooling change. The Control Decider takes the amount of cooling determined by the Cooling Decider and parcels out those changes to the controls in the order determined by the Priority Decider. In one implementation, this decision-making happens at polled intervals.

[00115] In one embodiment of the present invention, thermal manager 901 employs Fuzzy Logic and other data-driven algorithms to manage system temperature. For example, the Cooling Decider uses fuzzy logic principles and inference rules to determine the amount of cooling change, instead of modeling the system mathematically, in which the fuzzy logic model is empirically derived and modified through testing or simulation. An example of a rule is:

[00116] IF (temperature is hot) AND (temperature is increasing) THEN (turn cooling up).

[00117] After a number of rules are evaluated, their results are combined to generate a single result. Terms like (temperature is very hot), (temperature is increasing) or (turn cooling up a lot) can be defined for better precision in control.

[00118] For example, a Cooling Decider may use the following rules.

[00119] 1. IF (temperature is cold) THEN (turn cooling down)

[00120] 2. IF (temperature is warm) THEN (do nothing)

[00121] 3. IF (temperature is hot) THEN (turn cooling up)

[00122] **Figure 10** illustrates an example of a method to determine actions to be performed using fuzzy logic in operating a data processing system according to one embodiment of the present invention. A sensed temperature may be classified non-exclusively as cold, warm and hot. For example, if the current temperature is a number of degrees below the desired temperature, it can be classified mostly warm and a little cold. In **Figure 10**, membership functions 1001, 1003 and 1005 define the levels of truth for the classification of different temperatures. For example, curve 1001 represents the level of truth for different temperatures. When the different between the current temperature and the target temperature, $T-T_o$, is between -10^o and 0^o, the truth value increases linearly as the current temperature reduces (and decreases as the current temperature increases). When the current temperature is 10^o below the target temperature, the truth value of being cold is 1.0; and, when the current temperature is above the target temperature, the truth value of being cold is 0.0. Curve 1003 defines non-constant truth values of warm when the difference between the current temperature and the target temperature ($|T-T_o|$) is less than 10^o; and, curve 1005 defines the linear variation of truth values of hot when the current temperature is within 10^o above the target temperature. Thus, if a current temperature is 7.5^o, the truth values for cold, warm and hot are 0, 0.25 and 0.75 (1021, 1023 and 1025) respectively. The above rules for the cooling decider then lead (1011, 1013 and 1015) to the corresponding truth values 0, 0.25 and 0.75 for the actions (turn cooling down), (do nothing) and (turn cooling up) respectively, when the above inference rules are used.

[00123] Membership graphs may be of complex shapes, such as Gaussian curves. Keeping them to triangles and trapezoids makes the calculations much faster.

[00124] **Figures 11** and **12** illustrate an example defuzzification method to merge different actions as one quantified action to operate a data processing system according to one embodiment of the present invention.

[00125] To merge the different results, the Cooling Decider goes through a process called "defuzzification" to get a single, crisp result in the range of ±20 units for the cooling, assuming the change of cooling is always limited within 20 units. Consider an example in which the actions of (turn cooling down), (do nothing) and (turn cooling up) have member functions 1111, 1113 and 1115 respectfully, as shown in **Figure 11**. In **Figure 11**, when cooling is turned down by no more than 10 units, the truth value of (turn cooling down) increases linearly as the unit of cooling decreases; when cooling is turned up by no more than 10 units, the truth value of (turn cooling up) increases linearly as the unit of cooling increases; and, when the change in cooling ($|\Delta C|$) is less than 10 units, (Do nothing) has non-constant truth value.

[00126] Note that the shapes, ranges and slopes of member functions 1111, 113 and 1115 for (turn cooling down), (do nothing) and (turn cooling up) are in general different from those for cold, warm and hot.

[00127] A commonly used method for "defuzzification", called the Centroid algorithm, first clips each consequent (result) by the degree of truth of its antecedent. Since (turn cooling down), (do nothing) and (turn cooling up) have truth values 0, 0.25, 0.75 respectively, member functions 1111, 1113 and 1115 are clipped to generate functions 1101, 1103 and 1105 respectively.

[00128] Next, the clipped member functions are overlaid as in **Figure 12** to calculate an average point where there is an equal area under the graph on each side of the average point. This is analogous to finding the center of mass in physics and is called the Centroid Method. Curve 1203 in **Figure 12** corresponds to the portion of curve 1103 between points 1122 and 1125 in **Figure** 11; and, curve 1205 corresponds to the portion of curve 1105 beyond point 1126. The average point 1211 in **Figure 12** is at 10.2 unit. Thus, the Cooling Decider reaches the conclusion to turn up cooling for 10.2 units.

[00129] The priority decider and the control decider then determine how to turn up cooling for 10.2 units.

[00130] There are at least two kinds of data that are taken into account in deciding the priority ordering of controls: thermal zone and control type. For example, a system might have two fans: a CPU near one fan, and a GPU near the other. In this example, there might be two thermal zones, one specifying the CPU and its associated fan, and the other specifying the GPU and its fan. However, the fans, CPU, and GPU are different types of thermal controls: they create different side effects when in modulating the temperatures. Therefore, they are associated by type of control (e.g., fan, processor, etc). Each type of control has certain known properties, depending on the user's preferences or current environmental factors.

[00131] The Priority Decider prioritizes the controls by type and zone into a single priority queue for the Control Decider to adjust. For example, the list can be determined by user profiles, like "quiet" and "high performance." These profiles sort the controls by type. Within a type, the controls can be ordered by proximity to hot sensors: if a particular control is closer to the hot sensor than another similarly-typed control, it will have a higher priority. However, differently-typed controls are sorted according to profiles, not zones.

[00132] In one embodiment of the present invention, the Priority Decider is based on an expert system, which takes in dynamic system information and applies it to a list of rules to determine the priorities of the controls. The rules for the Priority Decider include system environmental rules and user preference rules. A level of priority indicates that the level of importance of the work of the device. Examples of system environmental rules include:

[00133] If intake temperature is high, decrease the fan's priority

[00134] If battery is charging and is above 90%, increase the battery charger's priority

[00135] If CPU load is low, decrease the CPU's priority

[00136] If zone x is hot, use controls in zone x first

[00137] If graphics pipeline is busy, decrease the GPU's priority

[00138] If DVD is playing, decrease the CPU's priority

[00139] If burning CD/DVD, decrease the CPU's priority

[00140] If filesystem is busy, decrease the hard drive's priority

[00141] If sensors are too hot, don't burn CD/DVD

[00142] If CPU speed is too slow, can't burn CD/DVD

[00143] Examples of user preference rules include:

[00144] If "quiet," decrease the fan's priority

[00145] If "high performance," decrease the CPU's priority

[00146] If "high performance," decrease the GPU's priority

[00147] The Control Decider adjusts the list of controls to achieve the required cooling change determined by the Cooling Decider, according to a list of controls ranked by the Priority Decider in the order they should be changed. In one embodiment, the cooling change that the Cooling Decider provides can be considered as the quantity of total control index change to be made, with 0 being no cooling change, positive representing more aggressive cooling, and negative representing less aggressive cooling.

[00148] For example, if the Cooling Decider determines to increase cooling for a number of units, the control with the lowest priority is adjusted first (e.g., to reach the maximum cooling capacity if necessary) to provide the required cooling. If the required unit of cooling is not satisfied after the total cooling capacity of the control with the lowest priority is exhausted, the control with the next lowest priority is adjusted. Thus, the list of controls is processed in the ascending order of priority for adjustment until the required units of cooling is satisfied.

[00149] Similarly, if the Cooling Decider determines to decrease cooling for a

number of units, the control with the highest priority is adjusted first (e.g., to reach the minimum cooling capacity and maximum performance) to accommodate the decrease of cooling. If an extra number of unit of cooling is available for decreasing after the cooling provided by the control with the highest priority reaches the minimum, the control with the next highest priority is adjusted. Thus, the list of controls is processed in the descending order of priority for adjustment until the given units of cooling is decreased.

[00150] In one embodiment of the present invention, the number of cooling units to be changed can be zero, with the changed priorities of the controls. To reflect the changes in priorities, the cooling units may be traded between controls of different priorities to achieve better performance. For example, if high priority control A is providing more cooling than low priority control B, the high priority control A is adjusted for higher performance but less cooling, while the low priority control B is adjusted to provide more cooling to compensate for the reduced cooling from control A.

[00151] In one embodiment of the present invention, the information about sensors and controls in the system (e.g., type, zone, and others), environmental parameters (e.g., high/low temperature thresholds), membership functions and inference rules required by the Cooling Decider, priority lists of controls needed for various user-affected settings, and others are collected (e.g., from device tree 913, Boot ROM 915) during an initialization period (e.g., at startup) of the management system.

[00152] **Figure 13** shows a software module diagram which shows software to manage the operation state of a data processing system according to one embodiment of the present invention. In one embodiment of the present invention, thermal manager manages system temperature through making simple decisions based on the temperature sensors and adjusting the CPU and GPU performance accordingly. In

one embodiment, the thermal manager is a platform dependent driver (e.g., platform monitor 1321) that responds to events generated by sensor drivers (e.g., in response to excessive thermal loading), power management requests (e.g., from power management 1313), or configuration changes from the user to modify the behavior of the system (e.g., by adjusting the working states of the components of the data processing system through controls 1325, 1327 and 1329). The thermal manager monitors the environmental factors (e.g., using sensors 1323) and takes necessary action to prevent damage to machine components or loss of user data.

[00153] In one embodiment of the present invention, state watcher 1311 allows a user to monitor sensors and observe the behavior of the thermal manager. This tool may be used to report all relevant data from the thermal management system, as well as make runtime tweaks to parameters in the system. For example, the user can see what the system would do if it were at a certain temperature or a certain state and set thresholds and polling periods for the individual sensors.

[00154] In one implementation, platform monitor 1321 implements a state machine, which in response to the input, adjusts various system parameters in order to adjust the level of cooling needed for the computer system. The knowledge about the temperature being managed and what constitutes too hot versus too cold are coded in the platform monitor. In one embodiment of the present invention, the platform monitor determines the current state of the system from the information obtained from the sensor drivers and adjusts the thermal controls of the system to move the system from one state to another, if the current state is not a desirable one. For example, the states are determined from the information collected from sensors (e.g., temperatures sensors), system conditions (e.g., lid open or closed), and preferences (e.g., "quiet" or "high performance"). There may also be different profiles for lid-closed, "quiet", or other situations so that the monitor actively manages the system to move between the states in the current profile. The platform

monitor is platform specific and has detailed knowledge of the platform on which it is running. Using this knowledge, it collects information from sensors in the system and takes appropriate actions based on the states of these sensors.

[00155] Sensor drivers provide environmental information about the computer. An instance of a sensor driver represents a specific environmental-sensing device, like a thermistor, an ambient light sensor, or a software sensor like kernel load factor. For example, temperature sensors provide temperature information about the data processing system on which the sensor drivers are running. A sensor driver gets information about the sensor (e.g., thermal zone, type (such as temperature, light, battery, kernel load), thresholds, and others). Sensor drivers can be polled to get the current values and may support event notification to signal important threshold conditions.

[00156] A sensor driver (e.g., 907 or 1323) may apply to any sensor that detects one aspect of the state of the system. For example, the user selecting "Reduced performance" in the Energy Saver preference panel is sensor input. Whether the user has open or closed the clamshell is sensed information about the environment of the data processing system. Any input may trigger a response that dictates a state change of the system; and, the platform monitor is the centralized decision maker for taking actions to adjust the state of the system.

[00157] In one embodiment, each sensor registers itself with the manager (e.g., platform monitor 1321 or thermal manager 901). For some sensors, such as temperature sensors, the manager sets a threshold that controls when the sensor driver notifies the manager. For example, a temperature sensor may be given an upper threshold and a lower threshold so that the sensor driver notifies the manager whenever the sensed temperature crosses one of those thresholds. The sensor may be a smart one so that it triggers an interrupt when a threshold is crossed. Alternatively, the sensor driver may simulate this behavior by polling the sensor and only notifying

also registers as a client of the sensor driver to establish communication. The sensor driver sends a message to the manager when an interesting event occurs, such as a threshold that has been hit or exceeded, or when the manager polls the sensor for the current value. In one embodiment of the present invention, the generic temperature sensor driver is a generic liaison driver. It obtains the actual value for the sensor by talking to a specific device driver.

[00163] Control drivers are the actual effectors of the state change. Examples of these include drivers that can change CPU multiplier, system bus speed, or GPU performance level. In one implementation, some of these controls are linked into the manager (e.g., platform monitor).

[00164] In one embodiment of the present invention, an instance of a control driver represents a device that is able to adjust its working state for environmental variation, including devices that are designed for removing heat, such as fans, or devices that can adjust their performance to reduce heat generation. Control drivers may be visualized as a dial for the output of the device they're controlling. In one implementation, a control driver accepts a value for the dial (e.g., indicating a working state as a index value within 0-100, with higher values representing most aggressive cooling and lower values representing least aggressive cooling); and, the control driver also supports reporting its current index value and information about whether it is at the maximum or minimum control level.

[00165] The devices that can have a corresponding control driver include: CPU, GPU, fan, backlight, battery charging, hard drive, optical drive, PCI card, and others. CPU and GPU can be of a type of "performance hit" for which different working states correspond to different tradeoff in performance and heat generation (or power consumption); fan can be of a type of "noisy" for which different working states correspond to different tradeoff in heat removal (and noise) and power consumption; backlight and battery charging can be of a type of "user impact" for which different working states correspond to different tradeoff

in user experience impact and power consumption.

[00166] The CPU clock is generated for some CPUs via an on-chip PLL that selectively multiplies and divides the processor bus clock. However, the CPU clock PLL configuration of many microprocessors is only programmable during a reset cycle. Thus, these CPU may be rebooted in order to change clock speeds. The special reset cycle for changing clock may be accomplished by programming a register in the memory controller for state initialization and then sending a command to the PMU for a reset.

[00167] The latency for switching the CPU multiplier is very high when a complete reset of the CPU is performed. In such an implementation, interrupts can be deferred for a period of time until the CPU reset is complete. This high interrupt latency can be evident to the end use in the form of audio drop outs and/or pops. Thus, such a method for CPU speed switch is generally not transparent without affecting the user experience. In one embodiment of the present invention, such an approach for CPU multiplier shift is used primarily to cope with excessive thermal stress.

[00168] To achieve a more transparent CPU speed change, the CPU clock is altered through slewing (slowly changing) the bus clock. Some microprocessors derive their clock from the system bus clock using an on-chip PLL. In one embodiment of the present invention, the processor clock is changed, without a reset, through slowly changing the processor bus clock at a rate slow enough to allow the on-chip PLL remain locked. One technical challenge is that changing the bus clock has the side effect of changing the rate at which the decrementer register is modified. In one embodiment of the present invention, several hardware components are used to implement CPU speed adjustment through slewing the bus clock, including a programmable clock source (e.g., Cypress CY28512) to support slewing and a circuitry in the core logic to handle the time base drift problem. Some CPUs have a signal called "TBEN" (Time Base ENable), which can be used to temporarily stop the processor from keeping track

of time. Custom logic inside the core logic modulates (changes) the TBEN signal in response to monitoring the master clocks. As a result, the CPU concept of time is updated at a constant rate, even though the bus clock changes with time.

[00169] In one embodiment of the present invention, a chip (e.g., CY28512) is used to take in a clock signal, apply some user-programmable multipliers to it, and output the clock signal at the resulting frequency. The formula that the chip uses to calculate the output frequency is as follows:

[00170] Output Frequency = Input frequency * (N / M)

[00171] The N and M values (along with some other options) are user programmable. For better usability, some chips (e.g., CY28512) accept two separate pairs of N and M values, and provides a way to switch between them. At initialization time, one pair is programmed with a set of "low" multipliers for the generation of a low output frequency, and the other pair with a set of "high" multipliers for the generation of a high output frequency. While the system is running, a control driver can toggle between them dynamically to effectively turn the clock frequency up and down. To slew the frequency slowly enough so that the on-chip PLL of the CPU can follow the frequency change, a number of frequency changes can be performed in small steps.

[00172] One advantage of lowering the clock frequency is that it allows the system to run at a lower voltage than normal to save power and reduce heat. Thus, after turning the clock down, the control driver can also turn the voltage down.

[00173] Some Graphics Processing Units (GPU) (e.g., the nVidia GeForce4Go (NV17M)) have a variety of power saving features designed (e.g., with a mobile application in mind). Some of these features are automatic; and, others are manually settable. For example, NV17M allows the chip to turn off unneeded areas to save power when they are not being used. Unused blocks can power down, as they are not needed. For example, the driver to an unattached display can be powered off if a second display is not attached, or the MPEG decoder cell

turned off if there is no need for it. The hardware clock saves power during tiny fractions of a second when the graphics hardware is not being used to its fullest.

[00174] The configurable feature of the NV17M is the ability to modify the swap interval. The swap interval defines the maximum number of frames the GPU renders per second. It also defines the number of screen refreshes between redraws. The overall effect is a change in GPU workload, which may also change the workload of the CPU. For example, the GPU may be configured to work at swap interval of 0 without power saving, at swap interval of 1 to generate frames at no faster than the display refresh rate, or at swap interval of 2 to generate frames at half the display refresh rate. In one embodiment of the present invention, the configurable swap interval is manipulated to limit temperature.

[00175] In one embodiment of the present invention, devices that act as sensors or controls are described in the device tree with a set of properties added to the nodes in the device tree, such as the type of sensor (i.e. temperature), a unique sensor ID, and the thermal zone of the sensor, location, polling period, and others. During the initialization period, control drivers and sensor drivers obtain this information from the device tree. For example, a control driver gets from the device tree information about the control, such as thermal zone, attributes (e.g., performance hit, noisy, cooling device), type of control (e.g., fan, processor, etc). The manager uses this information in determining the instantaneous index value for each control when new sensor values arrive. When an instantaneous index is received from the manager, the control driver communicates with one or more device drivers to adjust the control as necessary.

[00176] A state manager connects the sensor drivers and the control drivers to manage the working states of the components to provide the desirable result. For example, the levels of power and temperature dynamics may be control by the manager to best perform the current task within power and thermal constraints. The manager chooses the most relevant thermal controls to adjust, as well as how much to adjust. When some sensors indicate that they are at a particularly low

temperature, the polling of these sensors may be stopped after setting a threshold at which the sensors notify the system to restart polling them.

[00177] While the modules are functionally very independent, they may be in separate drivers or combined drivers. For example, some hardware device, such as fan controllers, may a combined sensor and control driver. If the only temperature of interest is the highest one of a number of thermistors, they may be combined into a single sensor. Further, the Cooling, Priority and Control Deciders may be just sets of routines within a single code module. From this description, a person skilled in the art can envision many different combinations, modifications and variations.

[00178] **Figures 14 – 16** show methods to operate a data processing system according to embodiments of the present invention.

[00179] In **Figure 14**, operation 1401 receives sensed information from a plurality of physical sensors (e.g., thermistor, tachometer) instrumented in a housing of a data processing system. Operation 1403 receives load information on processing loads (e.g., CPU load). Operation 1405 controls working states of a plurality of components of the data processing system in the housing according to the sensed information and the load information. The sensed information and the load information can be used to fine grain control the components to balance different goals, such as high performance, low power consumption, low acoustic noise, thermal constraints, user preferences, system design constraints, and others.

[00180] In **Figure 15**, after operation 1501 collects sensed information (e.g., CPU load and processor temperatures) from a plurality of sensors of a data processing system, operation 1503 determines a current state of the data processing system based on the sensed information and user preferences. Operation 1505 determines a target state according to a predetermined state diagram. Operation 1507 selectively adjusts a set of controls to change working states of components of the data processing system to move the system from the

current state to the target state. The state diagram may be pre-designed to specify control adjustments for transition from one state to another to balance different goals.

[00181] **Figure 18** illustrates an example of a state diagram which shows a way to operate a data processing system according to one embodiment of the present invention. In one embodiment of the present invention, the state of the system is determined from the temperature and the position of the lid. When the temperature is in the normal range, the system is either in state 1817 if the lid is in the open position or in state 1827 if the lid is in the closed position. When the system is in state 1817 or 1827, the system is allowed to operate at a maximum performance level. For example, the cooling provided by the CPU is at the lowest level (e.g., 0%), allowing a fast dynamic speed and the maximum computation performance; and, the cooling provided by the GPU is also at the lowest level (e.g., 0%). It is understood that the cooling provided by a processor (e.g., CPU or GPU) can be achieved through adjusting the working state of the processor to reduce the power consumption and the associated heat (e.g., through reducing the clock frequency and the core voltage). However, in one embodiment, when the lid is closed, the PMU is forced to run at a slow speed. As the temperature T increases to pass thresholds $T^a_{lid\text{-}open}$, $T^b_{lid\text{-}open}$, $T^c_{lid\text{-}open}$, and $T^d_{lid\text{-}open}$, the state of the system transits from state 1817 (normal) to states 1815 (warm), 1813 (very warm), 1811 (hot), and 1803 (very hot), if the lid is open. Similarly, as the temperature T increases to pass thresholds $T^a_{lid\text{-}closed}$, $T^b_{lid\text{-}closed}$, $T^c_{lid\text{-}closed}$, and $T^d_{lid\text{-}closed}$, the state of the system transits from state 1827 (normal, lid closed) to states 1825 (warm, lid closed), 1823 (very warm, lid closed), 1821 (hot, lid closed), and 1803 (very hot, lid closed), if the lid is closed. As the system goes into the states of high temperatures, the working states of the components of the system may be adjusted to cool down the system. For example, when in state 1815 (1825, 1813, 1823, 1811 or 1821), the cooling of the CPU may be adjusted to a higher level (e.g., 50%) to trade performance for cooling. For example, the

CPU may be forced into a slow dynamic speed. Further, when in a very hot state (1803), the working state of the CPU may be adjusted to provide maximum cooling (e.g., 100%). Similarly, when in state 1813 (or 1823), the cooling of the GPU may be adjusted to a higher level (e.g., 50%) to trade performance for cooling; and, when in state 1811 (1821, or 1803), the working state of the GPU may be adjusted to provide maximum cooling (e.g., 100%). To provide cooling from the graphics system, DVD (or other optical drive) speed may be reduced. When the temperature exceeds the safety threshold (e.g., $T > T_{safety}$), the system moves into a too hot state (1801), in which state PMU initiates a request for forced sleep. If the system remains in the too hot state (e.g., for 4 minutes) without going to a sleep mode, PMU triggers a forced shutdown. Without changing the temperature range, the system may transit between a lid closed state (e.g., 1821 – 1827) and a corresponding lid open state (e.g., 1811 – 1817) when the lid is opened or closed. When the system cools down, working states of the components (e.g., CPU, GPU, DVD) can be adjusted for higher performance. In **Figure 18**, different thresholds are used for defining the transition between two states due to the change in temperature. For example, when $T > T^{a}_{lid\text{-}open}$, the system moves from state 1817 (normal) to state 1815 (warm); and, the system moves back to state 1817 (normal) from state 1815 (warm) only when $T < T^{a}_{lid\text{-}open} - T_{hysterisis}$. The difference in the threshold, $T_{hysterisis}$, allows the system to be at one state when the temperature fluctuates only slightly near one threshold, avoiding unnecessary actions in adjusting working states.

[00182] Although one embodiment of the present invention uses the state diagram illustrated in **Figure 18** and operations for cooling adjustment for various states described above with **Figure 18**, it is understood that various different states of a state diagram can be defined and used for the operation of a computer system. Further, different transition paths and different adjustments of working states to more or less components (e.g., fan, memory chips, microprocessors, graphics chips, hard drives, optical drives, bridge chips, and

others) for cooling and performances can be defined for different state diagrams for operating a data processing system. For example, in one implementation, when T exceeds T_0, cooling fan is activated (e.g., 33% duty cycle for ADM103x); when $T \leq T_1$, CPU can run at full speed; when T exceeds T_2, cooling fan runs at full speed; when T exceeds T_3, CPU is forced into reduced speed mode; when T exceeds T_4, the system is forced to sleep, or to shutdown if not responding to the request to sleep.

[00183] In **Figure 16**, after operation 1601 collects sensed information (e.g., CPU load and processor temperatures) from a plurality of sensors of a data processing system, operation 1603 determines an amount of cooling change based on the sensed information. For example, fuzzy logic principles and inference rules can be used to determine the amount of cooling changes based on the sensed information. Operation 1605 determines a prioritized list of controls to balance different goals (e.g., performance, power consumption, thermal constraint, acoustic noise, user preference, system constraint). For example, an expert system can be used to prioritize the list according to a number of system rules and user preferences. Operation 1607 selectively adjusts a subset of the prioritized list of controls to effect the amount of cooling change. The amount of cooling change can be parceled out to one or more controls according to the priorities of the controls.

[00184] **Figure 17** illustrates a method to parcel out cooling changes to a number of controls according to one embodiment of the present invention.

[00185] If operation 1701 determines to increase cooling, operation 1711 first increases the speed of the cooling fan (e.g., up to the maximum fan speed when necessary). If operation 1713 determines that more cooling is required, operation 1715 decreases the CPU clock frequency within the allowable frequency range; and, operation 1717 decreases the CPU core voltage within the allowable frequency range. This adjustment to the CPU reduces the heat generation at the expense of computation performance. Thus, the thermal constraint can be

maintained while running the system at high computation performance.

[00186] If operation 1703 determines to decrease cooling, operation 1721 increases the CPU core voltage within the allowable voltage range; and, operation 1723 increases the CPU clock frequency within the allowable frequency range. This adjustment to the CPU increases the computation performance of the CPU and heat generation, which corresponds to decrease cooling. If the CPU is at the maximum performance state and operation 1725 determines less cooling is allowable, operation 1727 decreases the speed of the cooling fan to reduce noise and power consumption.

[00187] If operation 1731 determines that the CPU and the fan can trade cooling, operation 1733 increases the CPU core voltage; operation 1735 increases the CPU clock frequency; and, operation 1737 increases the speed of the cooling fan. Thus, the CPU is allowed to run at high performance, generating more heat, which is removed by increased cooling from the fan.

[00188] Thus, in **Figure 17**, cooling efforts are parceled out between the CPU and the cooling fan to have a high performance within a thermal constraint. In general, the cooling efforts can be parceled out (e.g., by a Control Decider) among a list of controls according to priorities (e.g., as determined by a Priority Decider), which reflect the balancing of different goals, such as performance, power consumption, thermal constraint, acoustic noise, user preference, system constraint).

[00189] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method to operate a data processing system, the method comprising:

determining a control level for a first component of the data processing system based on information obtained from a plurality of sensors, at least one of the sensors determining a temperature in the data processing system; and

automatically adjusting control of the first component according to the control level to move the first component from a first working state to a second working state.

2. A method as in claim 1, wherein the first component comprises a cooling fan of the data processing system; and, the cooling fan runs at a first speed in the first working state and a second speed in the second working state.

3. A method as in claim 2, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

4. A method as in claim 1, wherein the first component comprises a processor.

5. A method as in claim 4, wherein the first working state comprises a first clock frequency and a first core voltage for the processor; and, the second working state comprises a second clock frequency and a second core voltage for the processor.

6. A method as in claim 4, wherein the processor comprises a Graphics Processing Unit (GPU); the first working state comprises

a first swap interval; and, the second working state comprises a second swap interval.

7. A method as in claim 1, further comprising:

automatically adjusting control of a second component based on the information obtained from the plurality of sensors to move the second component from a third working state to a fourth working state.

8. A method as in claim 7, wherein the first component is a heat source of the data processing system and the second component is a cooling source of the data processing system.

9. A method as in claim 1, wherein the control level is determined further based on one or more user preferences.

10. A method as in claim 1, wherein one of the sensors comprises a software module determining a processor load of the data processing system.

11. A method to operate a data processing system, the method comprising:

determining a subset of controls from a plurality of controls of a plurality of components of the data processing system based on information obtained from a plurality of sensors, each of the plurality of sensors determining an aspect of a working condition of the data processing system; and

adjusting the subset of controls to change working states of corresponding components of the data processing system to balance requirements in performance and in at least one of: thermal constraint and power consumption.

12. A method as in claim 11, wherein the plurality of sensors comprise at least one of:

a) a temperature sensor;

b) a tachometer; and

c) a software module determining a load of a processor.

13. A method as in claim 11, wherein the plurality of components comprise heat sources of the data processing system and cooling sources of the data processing system.

14. A method as in claim 13, wherein the heat sources comprises at least one of:

a) a Central Processing Unit (CPU);

b) a Graphics Processing Unit (GPU);

c) a hard drive;

d) an optical drive; and

e) an Integrated Circuit (IC) chip.

15. A method as in claim 11, further comprising:

determining an amount of cooling change based on the information obtained from the plurality of sensors;

wherein the subset of controls are adjusted to effect the amount of cooling change.

16. A method as in claim 15, wherein the amount of cooling change is determined according to a fuzzy logic.

17. A method as in claim 16, wherein said determining the subset of controls comprises:

determining a prioritized list of the plurality of controls.

18. A method as in claim 17, wherein the prioritized list is determined at least partially based on one or more user preferences.

19. A method as in claim 17, further comprising:

parceling out the amount of cooling change to the subset of controls.

20. A method as in claim 11, further comprising:

determining a first state of the data processing system from the information obtained from the plurality of sensors;

wherein the subset of controls is determined from a decision to move the data processing system from the first state to a second state.

21. A method to operate a cooling fan of a data processing system, the method comprising:

adjusting the cooling fan from running at a first speed to running at a second speed in response to a temperature sensor measurement and a user preference.

22. A method as in claim 21, further comprising:

verifying that the cooling fan is running at the second speed.

23. A method as in claim 22, wherein tachometer information obtained from a fan controller for the cooling fan is used to verify that the cooling fan is running at the second speed.

24. A method as in claim 23, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

25. A method as in claim 21, further comprising:

determining one or more temperature measurements; and

determining the second speed for the cooling fan based at least partially on the one or more temperature measurements.

26. A method as in claim 25, wherein the one or more temperature measurements are obtained from one or more temperature sensors instrumented in the data processing system.

27. A method as in claim 26, wherein the one or more temperature measurements indicate temperatures of at least one of:

a) a microprocessor of the data processing system;

b) a graphics chip of the data processing system; and

c) a memory chip of the data processing system.

28. A method as in claim 27, wherein the microprocessor of the data processing system determines the second speed.

29. A method as in claim 25, wherein the second speed is determined further based on a user preference stored in a machine readable medium of the data processing system.

30. A method as in claim 25, wherein the second speed is determined further based on a computation load level on the data processing system.

31. A method to operate a processor of a data processing system, the method comprising:

shifting a power supply of the processor from a first voltage to a second voltage without resetting the processor.

32. A method as in claim 31, further comprising:

slewing a frequency of a clock of the data processing system to transit a clock of the processor from a first frequency to a second frequency.

33. A method as in claim 32, wherein the processor continues to execute instructions while the frequency of the clock is slewed.

34. A method as in claim 33, wherein the processor continues to execute instructions while the power supply is shifted from the first voltage to the second voltage.

35. A method as in claim 32, wherein the power supply is maintained at one of the first and second voltages while the frequency of the clock is slewed; and, the clock of the processor is maintained at one of the first and second frequencies while the power supply is shifted from the first voltage to the second voltage.

36. A method as in claim 32, wherein the first frequency is higher than the second frequency; the first voltage is higher than the second voltage; and, the power supply is shifted from the first voltage to

the second voltage after the clock of the processor transits from the first frequency to the second frequency.

37. A method as in claim 32, wherein the first frequency is lower than the second frequency; the first voltage is lower than the second voltage; and, the power supply is shifted from the first voltage to the second voltage before the clock of the processor transits from the first frequency to the second frequency.

38. A method as in claim 32, wherein said slewing a frequency of a clock comprises:

instructing a clock chip to use a new frequency multiplier.

39. A method as in claim 31, further comprising:

adjusting a frequency multiplier of the processor to switch a clock of the processor from a first frequency to a second frequency.

40. A method as in claim 39, wherein the processor is not reset during switching from the first frequency to the second frequency.

41. A data processing system, comprising:

a housing;

a plurality of components mounted within the housing, the plurality of components including:

a memory; and

a processor coupled to the memory; and

a plurality of sensors instrumented within the housing, at least one of the sensors determining a temperature in the data processing system, the processor determining a control level for a first component of the plurality of components based on information obtained from the plurality of sensors, the processor causing the first component be adjusted according to the control level to move the first component from a first working state to a second working state.

42. A data processing system as in claim 41, wherein the first component comprises a cooling fan of the data processing system; and, the cooling fan runs at a first speed in the first working state and a second speed in the second working state.

43. A data processing system as in claim 42, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

44. A data processing system as in claim 41, wherein the first component comprises the processor.

45. A data processing system as in claim 44, wherein the first working state comprises a first clock frequency and a first core voltage for the processor; and, the second working state comprises a second clock frequency and a second core voltage for the processor.

46. A data processing system as in claim 44, wherein the processor comprises a Graphics Processing Unit (GPU); the first working state comprises a first swap interval; and, the second working state comprises a second swap interval.

47. A data processing system as in claim 41, wherein the processor further cause a second component of the plurality of components be adjusted based on the information obtained from the plurality of sensors to move the second component from a third working state to a fourth working state.

48. A data processing system as in claim 47, wherein the first component is a heat source of the data processing system and the second component is a cooling source of the data processing system.

49. A data processing system as in claim 41, further comprising:

one or more input/output (I/O) devices coupled to the processor, the one or more input/output (I/O) devices receive one or more user preferences;

wherein the control level is determined further based on one or more user preferences.

50. A data processing system as in claim 41, wherein the processor determines a processor load of the data processing system; and, the control level is determined further based on the processor load.

51. A data processing system, comprising:

a housing;

a plurality of components mounted within the housing, the plurality of components including:

a memory; and

a processor coupled to the memory; and

a plurality of sensors instrumented within the housing, each of the plurality of sensors determining an aspect of a working condition of the data processing system, the processor determining a subset of controls from a plurality of controls of the plurality of components based on information obtained from the plurality of sensors, the processor adjusting the subset of controls to change working states of corresponding components of the data processing system to balance requirements in performance and in at least one of: thermal constraint and power consumption.

52. A data processing system as in claim 51, wherein the plurality of sensors comprise at least one of:

a) a temperature sensor; and

b) a tachometer.

53. A data processing system as in claim 51, wherein the plurality of components comprise heat sources of the data processing system and cooling sources of the data processing system.

54. A data processing system as in claim 53, wherein the heat sources comprises at least one of:

a) a Graphics Processing Unit (GPU);

b) a hard drive;

c) an optical drive; and

d) an Integrated Circuit (IC) chip.

55. A data processing system as in claim 51, wherein the processor determines an amount of cooling change based on the information obtained from the plurality of sensors; and, the subset of controls are adjusted to effect the amount of cooling change.

56. A data processing system as in claim 55, wherein the amount of cooling change is determined according to a fuzzy logic.

57. A data processing system as in claim 56, wherein the processor determines a prioritized list of the plurality of controls in determining the subset of controls

58. A data processing system as in claim 57, further comprising:

one or more input/output (I/O) devices coupled to the processor, the one or more input/output (I/O) devices receive one or more user preferences;

wherein the prioritized list is determined at least partially based on the one or more user preferences.

59. A data processing system as in claim 57, wherein the processor further parcels out the amount of cooling change to the subset of controls.

60. A data processing system as in claim 51, wherein the processor determines a first state of the data processing system from the

information obtained from the plurality of sensors; and, the subset of controls is determined from a decision to move the data processing system from the first state to a second state.

61. A data processing system, comprising:

a housing;

a cooling fan coupled with the housing;

a processor mounted within the housing, the processor causing the cooling fan from running at a first speed to running at a second speed in response to a temperature sensor measurement and a user preference.

62. A data processing system as in claim 61, further comprising:

a tachometer coupled with the cooling fan and coupled to the processor, the processor communicating with the tachometer to verify that the cooling fan is running at the second speed.

63. A data processing system as in claim 62, further comprising:

a fan controller coupled to the cooling fan and the processor, the the fan controller comprising the tachometer.

64. A data processing system as in claim 63, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

65. A data processing system as in claim 61, further comprising:

one or more temperature sensors instrumented within the house, the one or more temperature sensors determining one or more temperature measurements, the processor determining the second speed for the cooling fan based at least partially on the one or more temperature measurements.

66. A data processing system as in claim 65, further comprising:

a graphics chip coupled to the processor; and

a memory chip coupled to the processor.

67. A data processing system as in claim 66, wherein the one or more temperature measurements indicate temperatures of at least one of:

a) the processor;

b) the graphics chip; and

c) the memory chip.

68. A data processing system as in claim 65, further comprising:

one or more input/output (I/O) devices coupled to the processor;

memory coupled to the processor;

wherein the one or more I/O devices receive a user preference and the processor stores the user preference on the memory.

69. A data processing system as in claim 68, wherein the second speed is determined further based on the user preference.

70. A data processing system as in claim 65, wherein the processor determines a computation load level on the data processing system; and, the second speed is determined further based on the computation load level.

71. A data processing system, comprising:

a processor; and

a power supply coupled to the processor to energize the processor with a first voltage, the processor causing the power supply to adjust the first voltage to a second voltage without resetting the processor.

72. A data processing system as in claim 71, further comprising:

a clock source coupled to the processor, the processor deriving a clock of the processor from the clock source, the processor causing the clock source to slew a frequency of the clock source to transit the clock of the processor from a first frequency to a second frequency.

73. A data processing system as in claim 72, wherein the processor continues to execute instructions while the frequency of the clock source is slewed.

74. A data processing system as in claim 73, wherein the processor continues to execute instructions while the power supply is shifted from the first voltage to the second voltage.

75. A data processing system as in claim 72, wherein the power supply is maintained at one of the first and second voltages while the frequency of the clock source is slewed; and, the clock of the processor is maintained at one of the first and second frequencies while the power supply is shifted from the first voltage to the second voltage.

76. A data processing system as in claim 72, wherein the first frequency is higher than the second frequency; the first voltage is higher than the second voltage; and, the power supply is shifted from the first voltage to the second voltage after the clock of the processor transits from the first frequency to the second frequency.

77. A data processing system as in claim 72, wherein the first frequency is lower than the second frequency; the first voltage is lower than the second voltage; and, the power supply is shifted from the first voltage to the second voltage before the clock of the processor transits from the first frequency to the second frequency.

78. A data processing system as in claim 72, wherein the processor instructs the clock source to use a new frequency multiplier to slew the frequency of the clock source.

79. A data processing system as in claim 71, wherein the processor adjusts a frequency multiplier of the processor to switch a clock of the processor from a first frequency to a second frequency.

80. A data processing system as in claim 79, wherein the processor is not reset during switching from the first frequency to the second frequency.

81. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to operate the data processing system, the method comprising:

determining a control level for a first component of the data processing system based on information obtained from a plurality of sensors, at least one of the sensors determining a temperature in the data processing system; and

automatically adjusting control of the first component according to the control level to move the first component from a first working state to a second working state.

82. A medium as in claim 81, wherein the first component comprises a cooling fan of the data processing system; and, the cooling fan runs at a first speed in the first working state and a second speed in the second working state.

83. A medium as in claim 82, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

84. A medium as in claim 81, wherein the first component comprises a processor.

85. A medium as in claim 84, wherein the first working state comprises a first clock frequency and a first core voltage for the processor; and, the second working state comprises a second clock frequency and a second core voltage for the processor.

86. A medium as in claim 84, wherein the processor comprises a Graphics Processing Unit (GPU); the first working state comprises

a first swap interval; and, the second working state comprises a second swap interval.

87. A medium as in claim 81, wherein the method further comprises:

automatically adjusting control of a second component based on the information obtained from the plurality of sensors to move the second component from a third working state to a fourth working state.

88. A medium as in claim 87, wherein the first component is a heat source of the data processing system and the second component is a cooling source of the data processing system.

89. A medium as in claim 81, wherein the control level is determined further based on one or more user preferences.

90. A medium as in claim 81, wherein one of the sensors comprises a software module determining a processor load of the data processing system.

91. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to operate the data processing system, the method comprising:

determining a subset of controls from a plurality of controls of a plurality of components of the data processing system based on information obtained from a plurality of sensors, each of the plurality of sensors determining an aspect of a working condition of the data processing system; and

adjusting the subset of controls to change working states of corresponding components of the data processing system to balance requirements in performance and in at least one of: thermal constraint and power consumption.

92. A medium as in claim 91, wherein the plurality of sensors comprise at least one of:

a) a temperature sensor;

b) a tachometer; and

c) a software module determining a load of a processor.

93. A medium as in claim 91, wherein the plurality of components comprise heat sources of the data processing system and cooling sources of the data processing system.

94. A medium as in claim 93, wherein the heat sources comprises at least one of:

a) a Central Processing Unit (CPU);

b) a Graphics Processing Unit (GPU);

c) a hard drive;

d) an optical drive; and

e) an Integrated Circuit (IC) chip.

95. A medium as in claim 91, wherein the method further comprises:

determining an amount of cooling change based on the information obtained from the plurality of sensors;

wherein the subset of controls are adjusted to effect the amount of cooling change.

96. A medium as in claim 95, wherein the amount of cooling change is determined according to a fuzzy logic.

97. A medium as in claim 96, wherein said determining the subset of controls comprises:

determining a prioritized list of the plurality of controls.

98. A medium as in claim 97, wherein the prioritized list is determined at least partially based on one or more user preferences.

99. A medium as in claim 97, wherein the method further comprises:

parceling out the amount of cooling change to the subset of controls.

100. A medium as in claim 91, wherein the method further comprises:
determining a first state of the data processing system from the information obtained from the plurality of sensors;
wherein the subset of controls is determined from a decision to move the data processing system from the first state to a second state.

101. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to operate a cooling fan of the data processing system, the method comprising:
adjusting the cooling fan from running at a first speed to running at a second speed in response to a temperature sensor measurement and a user preference.

102. A medium as in claim 101, wherein the method further comprises:
verifying that the cooling fan is running at the second speed.

103. A medium as in claim 102, wherein tachometer information obtained from a fan controller for the cooling fan is used to verify that the cooling fan is running at the second speed.

104. A medium as in claim 103, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

105. A medium as in claim 101, wherein the method further comprises:
determining one or more temperature measurements; and
determining the second speed for the cooling fan based at least partially on the one or more temperature measurements.

106. A medium as in claim 105, wherein the one or more temperature measurements are obtained from one or more temperature sensors instrumented in the data processing system.

107. A medium as in claim 106, wherein the one or more temperature measurements indicate temperatures of at least one of:

a) a microprocessor of the data processing system;

b) a graphics chip of the data processing system; and

c) a memory chip of the data processing system.

108. A medium as in claim 107, wherein the microprocessor of the data processing system determines the second speed.

109. A medium as in claim 105, wherein the second speed is determined further based on a user preference stored in a machine readable medium of the data processing system.

110. A medium as in claim 105, wherein the second speed is determined further based on a computation load level on the data processing system.

111. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to operate a processor of the data processing system, the method comprising:

shifting a power supply of the processor from a first voltage to a second voltage without resetting the processor.

112. A medium as in claim 111, wherein the method further comprises:

slewing a frequency of a clock of the data processing system to transit a clock of the processor from a first frequency to a second frequency.

113. A medium as in claim 112, wherein the processor continues to execute instructions while the frequency of the clock is slewed.

114. A medium as in claim 113, wherein the processor continues to execute instructions while the power supply is shifted from the first voltage to the second voltage.

115. A medium as in claim 112, wherein the power supply is maintained at one of the first and second voltages while the frequency of the clock is slewed; and, the clock of the processor is

maintained at one of the first and second frequencies while the power supply is shifted from the first voltage to the second voltage.

116. A medium as in claim 112, wherein the first frequency is higher than the second frequency; the first voltage is higher than the second voltage; and, the power supply is shifted from the first voltage to the second voltage after the clock of the processor transits from the first frequency to the second frequency.

117. A medium as in claim 112, wherein the first frequency is lower than the second frequency; the first voltage is lower than the second voltage; and, the power supply is shifted from the first voltage to the second voltage before the clock of the processor transits from the first frequency to the second frequency.

118. A medium as in claim 112, wherein said slewing a frequency of a clock comprises:

instructing a clock chip to use a new frequency multiplier.

119. A medium as in claim 111, wherein the method further comprises:

adjusting a frequency multiplier of the processor to switch a clock of the processor from a first frequency to a second frequency.

120. A medium as in claim 119, wherein the processor is not reset during switching from the first frequency to the second frequency.

121. A data processing system, comprising:

means for determining a control level for a first component of the data processing system based on information obtained from a plurality of sensors, at least one of the sensors determining a temperature in the data processing system; and

means for automatically adjusting control of the first component according to the control level to move the first component from a first working state to a second working state.

122. A data processing system as in claim 121, wherein the first component comprises a cooling fan of the data processing system; and, the cooling fan runs at a first speed in the first working state and a second speed in the second working state.

123. A data processing system as in claim 122, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

124. A data processing system as in claim 121, wherein the first component comprises a processor.

125. A data processing system as in claim 124, wherein the first working state comprises a first clock frequency and a first core voltage for the processor; and, the second working state comprises a second clock frequency and a second core voltage for the processor.

126. A data processing system as in claim 124, wherein the processor comprises a Graphics Processing Unit (GPU); the first working state comprises a first swap interval; and, the second working state comprises a second swap interval.

127. A data processing system as in claim 121, further comprising:

means for automatically adjusting control of a second component based on the information obtained from the plurality of sensors to move the second component from a third working state to a fourth working state.

128. A data processing system as in claim 127, wherein the first component is a heat source of the data processing system and the second component is a cooling source of the data processing system.

129. A data processing system as in claim 121, wherein the control level is determined further based on one or more user preferences.

130. A data processing system as in claim 121, wherein one of the sensors comprises a software module determining a processor load of the data processing system.

131. A data processing system, comprising:

means for determining a subset of controls from a plurality of controls of a plurality of components of the data processing system based on information obtained from a plurality of sensors, each of the plurality of sensors determining an aspect of a working condition of the data processing system; and

means for adjusting the subset of controls to change working states of corresponding components of the data processing system to balance requirements in performance and in at least one of: thermal constraint and power consumption.

132. A data processing system as in claim 131, wherein the plurality of sensors comprise at least one of:

a) a temperature sensor;

b) a tachometer; and

c) a software module determining a load of a processor.

133. A data processing system as in claim 131, wherein the plurality of components comprise heat sources of the data processing system and cooling sources of the data processing system.

134. A data processing system as in claim 133, wherein the heat sources comprises at least one of:

a) a Central Processing Unit (CPU);

b) a Graphics Processing Unit (GPU);

c) a hard drive;

d) an optical drive; and

e) an Integrated Circuit (IC) chip.

135. A data processing system as in claim 131, further comprising:

means for determining an amount of cooling change based on the information obtained from the plurality of sensors;

wherein the subset of controls are adjusted to effect the amount of cooling change.

136. A data processing system as in claim 135, wherein the amount of cooling change is determined according to a fuzzy logic.

137. A data processing system as in claim 136, wherein said means for determining the subset of controls comprises:

means for determining a prioritized list of the plurality of controls.

138. A data processing system as in claim 137, wherein the prioritized list is determined at least partially based on one or more user preferences.

139. A data processing system as in claim 137, further comprising:

means for parceling out the amount of cooling change to the subset of controls.

140. A data processing system as in claim 131, further comprising:

means for determining a first state of the data processing system from the information obtained from the plurality of sensors;

wherein the subset of controls is determined from a decision to move the data processing system from the first state to a second state.

141. A data processing system, comprising:

a cooling fan; and

means for adjusting the cooling fan from running at a first speed to running at a second speed in response to a temperature sensor measurement and a user preference.

142. A data processing system as in claim 141, further comprising:

means for verifying that the cooling fan is running at the second speed.

143. A data processing system as in claim 142, wherein tachometer information obtained from a fan controller for the cooling fan is used to verify that the cooling fan is running at the second speed.

144. A data processing system as in claim 143, wherein a duty cycle of the cooling fan is adjusted to run the cooling fan from the first speed to the second speed.

145. A data processing system as in claim 141, further comprising:

means for determining one or more temperature measurements; and

means for determining the second speed for the cooling fan based at least partially on the one or more temperature measurements.

146. A data processing system as in claim 145, wherein the one or more temperature measurements are obtained from one or more temperature sensors instrumented in the data processing system.

147. A data processing system as in claim 146, wherein the one or more temperature measurements indicate temperatures of at least one of:

a) a microprocessor of the data processing system;

b) a graphics chip of the data processing system; and

c) a memory chip of the data processing system.

148. A data processing system as in claim 147, wherein the microprocessor of the data processing system determines the second speed.

149. A data processing system as in claim 145, wherein the second speed is determined further based on a user preference stored in a machine readable medium of the data processing system.

150. A data processing system as in claim 145, wherein the second speed is determined further based on a computation load level on the data processing system.

151. A data processing system, comprising:

a processor; and

means for shifting a power supply of the processor from a first voltage to a second voltage without resetting the processor.

152. A data processing system as in claim 151, further comprising:

means for slewing a frequency of a clock of the data processing system to transit a clock of the processor from a first frequency to a second frequency.

153. A data processing system as in claim 152, wherein the processor continues to execute instructions while the frequency of the clock is slewed.

154. A data processing system as in claim 153, wherein the processor continues to execute instructions while the power supply is shifted from the first voltage to the second voltage.

155. A data processing system as in claim 152, wherein the power supply is maintained at one of the first and second voltages while the frequency of the clock is slewed; and, the clock of the processor is maintained at one of the first and second frequencies while the power supply is shifted from the first voltage to the second voltage.

156. A data processing system as in claim 152, wherein the first frequency is higher than the second frequency; the first voltage is higher than the second voltage; and, the power supply is shifted from the first voltage to the second voltage after the clock of the processor transits from the first frequency to the second frequency.

157. A data processing system as in claim 152, wherein the first frequency is lower than the second frequency; the first voltage is lower than the second voltage; and, the power supply is shifted from the first voltage to the second voltage before the clock of the processor transits from the first frequency to the second frequency.

158. A data processing system as in claim 152, wherein said means for slewing a frequency of a clock comprises:

means for instructing a clock chip to use a new frequency multiplier.

159. A data processing system as in claim 151, further comprising:

means for adjusting a frequency multiplier of the processor to switch a clock of the processor from a first frequency to a second frequency.

160. A data processing system as in claim 159, wherein the processor is not reset during switching from the first frequency to the second frequency.

161. A data processing system, comprising:

a housing;

a plurality of components mounted within the housing, the plurality of components including:

a memory;

a processor coupled to the memory; and

a power management unit coupled to the processor; and

a plurality of sensors instrumented within the housing, at least one of the sensors determining a temperature in the data processing system, according to instructions stored in the memory the processor managing working states of at least a portion of the plurality of components based on information obtained from at least a portion of the plurality of sensors, the power management unit causing the system a shutdown when a measurement of one of the plurality of sensors exceeds a threshold.

162. A data processing system as in claim 161, wherein the power management unit sends a signal to the processor when the measurement exceeds the threshold.

163. A data processing system as in claim 162, wherein the processor instructs the system to shutdown in response to the signal from the power management unit.

164. A data processing system as in claim 162, wherein the power management unit determines whether or nor the processor is responsive to the signal.

165. A data processing system as in claim 164, wherein the power management unit shuts off power to at least a portion of the system in response to a determination that the processor is not responsive to the signal.

166. A data processing system as in claim 165, wherein the power management unit comprises a watchdog timer; the processor is not responsive to the signal if the watchdog timer expires after the signal is sent to the processor.

167. A data processing system as in claim 161, wherein information about the shutdown is recorded in the memory.

168. A data processing system as in claim 167, wherein the processor records the information about the shutdown and instructs the system to shutdown in response to a signal from the power management unit.

169. A data processing system as in claim 167, wherein the power management unit records the information about the shutdown.

170. A method performed by a data processing system, the method comprising:

automatically managing, according to instructions stored in a memory of the data processing system, working states of a plurality of components of the data processing system based on information obtained from at least a portion of a plurality of sensors instrumented within a housing of the data processing system, at

least one of the sensors determining a temperature in the data processing system; and

initiating a shutdown from a power management unit of the data processing system in response to a measurement of one of the plurality of sensors exceeding a threshold.

171. A method as in claim 170, wherein the power management unit sends a signal to a processor of the data processing system when the measurement exceeds the threshold.

172. A method as in claim 171, wherein the processor instructs the system to shutdown in response to the signal from the power management unit.

173. A method as in claim 171, wherein the power management unit determines whether or nor the processor is responsive to the signal.

174. A method as in claim 173, wherein the power management unit shuts off power to at least a portion of the system in response to a determination that the processor is not responsive to the signal.

175. A method as in claim 174, wherein the power management unit comprises a watchdog timer; the processor is not responsive to the signal if the watchdog timer expires after the signal is sent to the processor.

176. A method as in claim 170, wherein information about the shutdown is recorded in the memory.

177. A method as in claim 176, wherein a processor of the data processing system records the information about the shutdown and instructs the system to shutdown in response to a signal from the power management unit.

178. A method as in claim 176, wherein the power management unit records the information about the shutdown.

179. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method performed by a data processing system, the method comprising:

automatically managing working states of a plurality of components of the data processing system based on information obtained from at least a portion of a plurality of sensors instrumented within a housing of the data processing system, at least one of the sensors determining a temperature in the data processing system;

wherein a power management unit of the data processing system initiates a shutdown in response to a measurement of one of the plurality of sensors exceeding a threshold.

180. A medium as in claim 179, wherein the power management unit sends a signal to a processor of the data processing system when the measurement exceeds the threshold.

181. A medium as in claim 180, wherein the processor instructs the system to shutdown in response to the signal from the power management unit.

182. A medium as in claim 180, wherein the power management unit determines whether or nor the processor is responsive to the signal.

183. A medium as in claim 182, wherein the power management unit shuts off power to at least a portion of the system in response to a determination that the processor is not responsive to the signal.

184. A medium as in claim 183, wherein the power management unit comprises a watchdog timer; the processor is not responsive to the signal if the watchdog timer expires after the signal is sent to the processor.

185. A medium as in claim 179, wherein information about the shutdown is recorded in a memory of the data processing system.

186. A medium as in claim 185, wherein a processor of the data processing system records the information about the shutdown and instructs the system to shutdown in response to a signal from the power management unit.

187. A medium as in claim 185, wherein the power management unit records the information about the shutdown.

188. A data processing system, comprising:

means for automatically managing working states of a plurality of components of the data processing system based on information obtained from at least a portion of a plurality of sensors instrumented within a housing of the data processing system, at least one of the sensors determining a temperature in the data processing system; and

means for initiating a shutdown from a power management unit of the data processing system in response to a measurement of one of the plurality of sensors exceeding a threshold.

189. A data processing system as in claim 188, wherein the power management unit sends a signal to a processor of the data processing system when the measurement exceeds the threshold.

190. A data processing system as in claim 189, wherein the processor instructs the system to shutdown in response to the signal from the power management unit.

191. A data processing system as in claim 189, wherein the power management unit determines whether or nor the processor is responsive to the signal.

192. A data processing system as in claim 191, wherein the power management unit shuts off power to at least a portion of the

system in response to a determination that the processor is not responsive to the signal.

193. A data processing system as in claim 192, wherein the power management unit comprises a watchdog timer; the processor is not responsive to the signal if the watchdog timer expires after the signal is sent to the processor.

194. A data processing system as in claim 188, wherein information about the shutdown is recorded in the memory of the data processing system.

195. A data processing system as in claim 194, wherein a processor of the data processing system records the information about the shutdown and instructs the system to shutdown in response to a signal from the power management unit.

196. A data processing system as in claim 194, wherein the power management unit records the information about the shutdown.

197. A method to manage a data process system enclosed in an housing of the data processing system, the method comprising:

individually monitoring a plurality of temperatures in a plurality of thermal zones respectively, the plurality of thermal zones being within the housing, the plurality of thermal zones not being isolated from each other, each of the plurality of thermal zones comprising at least one component that generates heat to substantially influence one of the plurality of temperatures when performing operations in a corresponding one of the plurality of thermal zones, each of the plurality of thermal zones comprising at least one component adjustable to reduce heat in a corresponding one of the plurality of thermal zones; and

adjusting working states of components in the plurality of thermal zones separately according to the plurality of temperatures respectively

to limit measurements of the plurality of temperatures to allowable levels.

198. The method of claim 197, wherein the at least one component adjustable to reduce heat comprises a cooling fan; and the at least one component that generates heat comprises an Integrated Circuit (IC) chip.

199. The method of claim 198, wherein the measurements of the plurality of temperatures are obtained from one or more thermal diodes integrated on the IC chip; and the IC chip comprises one of:

a microprocessor;

a graphics chip; and

a microcontroller.

200. The method of claim 198, further comprising:

determining a control setting for the cooling fan, the control setting causing the cooling fan to run at a desired speed;

wherein said adjusting comprises:

adjusting control of the cooling fan to the control setting to cause the cooling fan running at a current speed to run at the desired speed.

201. The method of claim 200, further comprising:

verifying that the cooling fan is running at the desired speed.

202. The method of claim 200, wherein said adjusting further comprises:

slowly adjusting a clock source and a voltage source to cause the IC chip running at a current clock frequency and a current core voltage to run at a desired clock frequency and a desired core voltage.

203. A method to operate a data process system enclosed in an housing of the data processing system, the method comprising:

receiving measurements from a plurality of temperature sensors instrumented in the data processing system;

adjusting working states of components of the data processing system to limit the measurements to allowable levels; and

in response to one of the measurements exceeding an allowable level:

storing data indicating a cause of turning off the data processing system; and

automatically turning off the data processing system to prevent damage to the data processing system.

204. The method of claim 203, further comprising:

determining whether or not a heatsink of the data processing system is functioning properly.

205. The method of claim 204, wherein when the heatsink is misapplied, adjusting the working states cannot limit at least one of the measurements to an allowable level in a typical operating environment.

206. The method of claim 204, wherein the data indicates that a software process for adjusting working states of the components failed to respond.

207. The method of claim 203, further comprising:

informing a user of trading performance for reduced heat when one or more components are adjusted to a low performance working state.

208. The method of claim 207, wherein the low performance working state comprises running a processor at a reduced frequency and a reduced core voltage.

209. A data process system enclosed in an housing of the data processing system, the data processing system comprising:

memory to store instructions;

a processor coupled to the memory; and

a plurality of temperature sensors coupled to the processor, the plurality of temperature sensors to individually monitor a plurality of temperatures in a plurality of thermal zones respectively, the plurality of thermal zones being within the housing, the plurality of thermal zones not being isolated from each other, each of the plurality of thermal zones comprising at least one component that generates heat to substantially influence one of the plurality of temperatures when performing operations in a corresponding one of the plurality of thermal zones, each of the plurality of thermal zones comprising at least one component adjustable to reduce heat in a corresponding one of the plurality of thermal zones, the plurality of the temperature sensors to provide measures of the plurality of temperatures to the processor, according to the instructions the processor to adjust working states of components in the plurality of thermal zones separately according to the plurality of temperatures respectively to limit measurements of the plurality of temperatures to allowable levels.

210. The system of claim 209, wherein the at least one component adjustable to reduce heat comprises a cooling fan; and the at least one component that generates heat comprises an Integrated Circuit (IC) chip.

211. The system of claim 210, wherein the measurements of the plurality of temperatures are obtained from one or more thermal diodes integrated on the IC chip; and the IC chip comprises one of:

a microprocessor;

a graphics chip; and

a microcontroller.

212. The system of claim 210, wherein the processor is further configured to determine a control setting for the cooling fan, the

control setting causing the cooling fan to run at a desired speed; and wherein to adjust the working states of the components, the processor is configured to adjust control of the cooling fan to the control setting to cause the cooling fan running at a current speed to run at the desired speed.

213. The system of claim 212, further comprising:

a tachometer coupled to the cooling fan and to the processor, the processor retrieving measurements from the tachometer to verify that the cooling fan is running at the desired speed.

214. The system of claim 212, further comprising:

a adjustable clock source coupled to the IC chip;

a voltage source coupled to the IC chip; and

wherein to adjust the working states of the components, the processor is configured to slowly adjust the clock source and the voltage source to cause the IC chip running at a current clock frequency and a current core voltage to run at a desired clock frequency and a desired core voltage.

215. A data process system enclosed in an housing of the data processing system, the data processing system comprising:

memory to store instructions;

a processor coupled to the memory;

a plurality of temperature sensors instrumented in the data processing system and coupled to the processor, the processor to receive measurements from the plurality of temperature sensors, according to the instructions the processor to adjust working states of components of the data processing system to limit the measurements to allowable levels;

a power management unit coupled to the processor, in response to one of the measurements exceeding an allowable level:

the power management unit and the processor to store data indicating a cause of turning off the data processing system in the memory;

the power management unit to automatically turn off the data processing system to prevent damage to the data processing system.

216. The system of claim 215, wherein the processor is configured according to the instructions to determine whether or not a heatsink of the data processing system is functioning properly.

217. The system of claim 216, wherein when the heatsink is misapplied, adjusting the working states cannot limit at least one of the measurements to an allowable level in a typical operating environment.

218. The system of claim 216, wherein the data indicates that a software process for adjusting working states of the components failed to respond.

219. The system of claim 215, wherein the processor is configured according to the instructions to inform a user of trading performance for reduced heat when one or more components are adjusted to a low performance working state.

220. The system of claim 219, wherein the low performance working state comprises running a processor at a reduced frequency and a reduced core voltage.

221. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to manage the data process system enclosed in an

housing of the data processing system, the method comprising:

individually monitoring a plurality of temperatures in a plurality of thermal zones respectively, the plurality of thermal zones being within the housing, the plurality of thermal zones not being isolated from each other, each of the plurality of thermal zones comprising at least one component that generates heat to substantially influence one of the plurality of temperatures when performing operations in a corresponding one of the plurality of thermal zones, each of the plurality of thermal zones comprising at least one component adjustable to reduce heat in a corresponding one of the plurality of thermal zones; and

adjusting working states of components in the plurality of thermal zones separately according to the plurality of temperatures respectively to limit measurements of the plurality of temperatures to allowable levels.

222. The medium of claim 221, wherein the at least one component adjustable to reduce heat comprises a cooling fan; and the at least one component that generates heat comprises an Integrated Circuit (IC) chip.

223. The medium of claim 222, wherein the measurements of the plurality of temperatures are obtained from one or more thermal diodes integrated on the IC chip; and the IC chip comprises one of:

a microprocessor;

a graphics chip; and

a microcontroller.

224. The medium of claim 222, wherein the method further comprises:

determining a control setting for the cooling fan, the control setting causing the cooling fan to run at a desired speed;

wherein said adjusting comprises:

adjusting control of the cooling fan to the control setting to cause the cooling fan running at a current speed to run at the desired speed.

225. The medium of claim 224, wherein the method further comprises:

verifying that the cooling fan is running at the desired speed.

226. The medium of claim 224, wherein said adjusting further comprises:

slowly adjusting a clock source and a voltage source to cause the IC chip running at a current clock frequency and a current core voltage to run at a desired clock frequency and a desired core voltage.

227. A machine readable medium containing executable computer program instructions which when executed by a data processing system cause said system to perform a method to operate the data process system enclosed in an housing of the data processing system, the method comprising:

receiving measurements from a plurality of temperature sensors instrumented in the data processing system;

adjusting working states of components of the data processing system to limit the measurements to allowable levels; and

in response to one of the measurements exceeding an allowable level:

storing data indicating a cause of turning off the data processing system; and

automatically turning off the data processing system to prevent damage to the data processing system.

228. The medium of claim 227, wherein the method further comprises:

determining whether or not a heatsink of the data processing system is functioning properly.

229. The medium of claim 228, wherein when the heatsink is misapplied, adjusting the working states cannot limit at least one of the measurements to an allowable level in a typical operating environment.

230. The medium of claim 228, wherein the data indicates that a software process for adjusting working states of the components failed to respond.

231. The medium of claim 227, wherein the method further comprises:

informing a user of trading performance for reduced heat when one or more components are adjusted to a low performance working state.

232. The medium of claim 231, wherein the low performance working state comprises running a processor at a reduced frequency and a reduced core voltage.

233. A data process system enclosed in an housing of the data processing system, the data processing system comprising:

means for individually monitoring a plurality of temperatures in a plurality of thermal zones respectively, the plurality of thermal zones being within the housing, the plurality of thermal zones not being isolated from each other, each of the plurality of thermal zones comprising at least one component that generates heat to substantially influence one of the plurality of temperatures when performing operations in a corresponding one of the plurality of thermal zones, each of the plurality of thermal zones comprising at least one component adjustable to reduce heat in a corresponding one of the plurality of thermal zones; and

means for adjusting working states of components in the plurality of thermal zones separately according to the plurality of temperatures

respectively to limit measurements of the plurality of temperatures to allowable levels.

234. The system of claim 233, wherein the at least one component adjustable to reduce heat comprises a cooling fan; and the at least one component that generates heat comprises an Integrated Circuit (IC) chip.

235. The system of claim 234, wherein the measurements of the plurality of temperatures are obtained from one or more thermal diodes integrated on the IC chip; and the IC chip comprises one of:

a microprocessor;

a graphics chip; and

a microcontroller.

236. The system of claim 234, further comprising:

means for determining a control setting for the cooling fan, the control setting causing the cooling fan to run at a desired speed;

wherein said means for adjusting comprises:

means for adjusting control of the cooling fan to the control setting to cause the cooling fan running at a current speed to run at the desired speed.

237. The system of claim 236, further comprising:

means for verifying that the cooling fan is running at the desired speed.

238. The system of claim 236, wherein said means for adjusting further comprises:

means for slowly adjusting a clock source and a voltage source to cause the IC chip running at a current clock frequency and a current core voltage to run at a desired clock frequency and a desired core voltage.

239. A data process system enclosed in an housing of the data processing system, the data processing system comprising:

means for receiving measurements from a plurality of temperature sensors instrumented in the data processing system;

means for adjusting working states of components of the data processing system to limit the measurements to allowable levels; and

in response to one of the measurements exceeding an allowable level:

means for storing data indicating a cause of turning off the data processing system; and

means for automatically turning off the data processing system to prevent damage to the data processing system.

240. The system of claim 239, further comprising:

means for determining whether or not a heatsink of the data processing system is functioning properly.

241. The system of claim 240, wherein when the heatsink is misapplied, said means for adjusting the working states cannot limit at least one of the measurements to an allowable level in a typical operating environment.

242. The system of claim 240, wherein the data indicates that a software process for adjusting working states of the components failed to respond.

243. The system of claim 239, further comprising:

means for informing a user of trading performance for reduced heat when one or more components are adjusted to a low performance working state.

244. The system of claim 243, wherein the low performance working state comprises running a processor at a reduced frequency and a reduced core voltage.

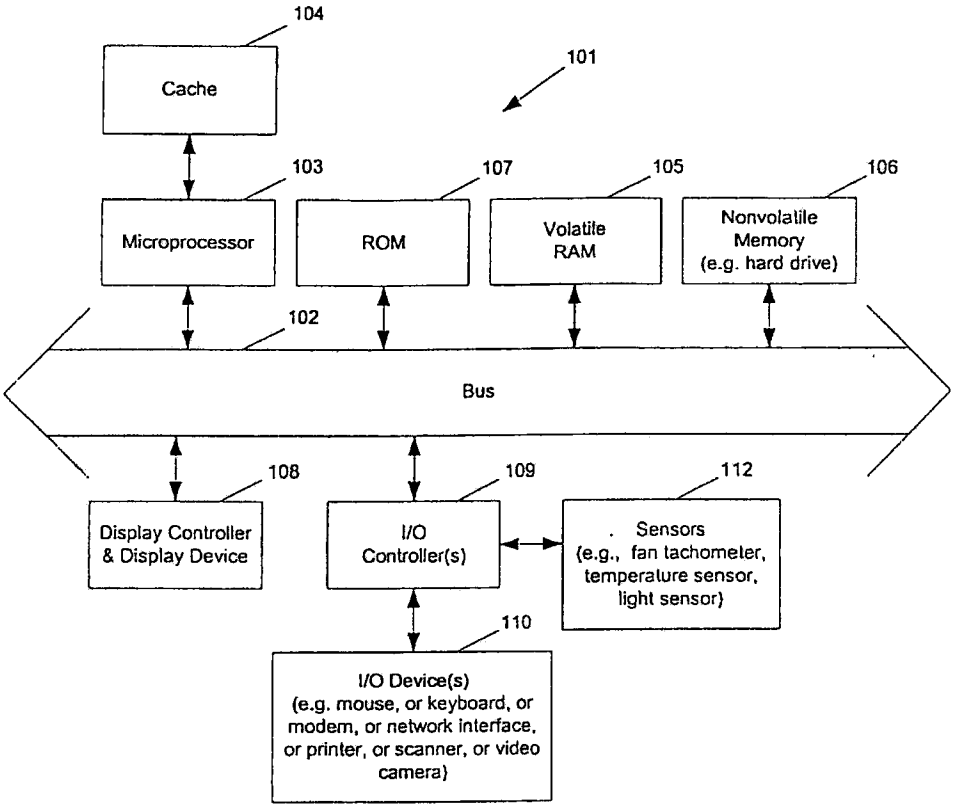

104
Cache
101
103
107
105
106
Microprocessor
ROM
Volatile RAM
Nonvolatile Memory (e.g. hard drive)
102
Bus
108
109
112
Display Controller & Display Device
I/O Controller(s)
Sensors (e.g., fan tachometer, temperature sensor, light sensor)
110
I/O Device(s) (e.g. mouse, or keyboard, or modem, or network interface, or printer, or scanner, or video camera)

Fig. 1

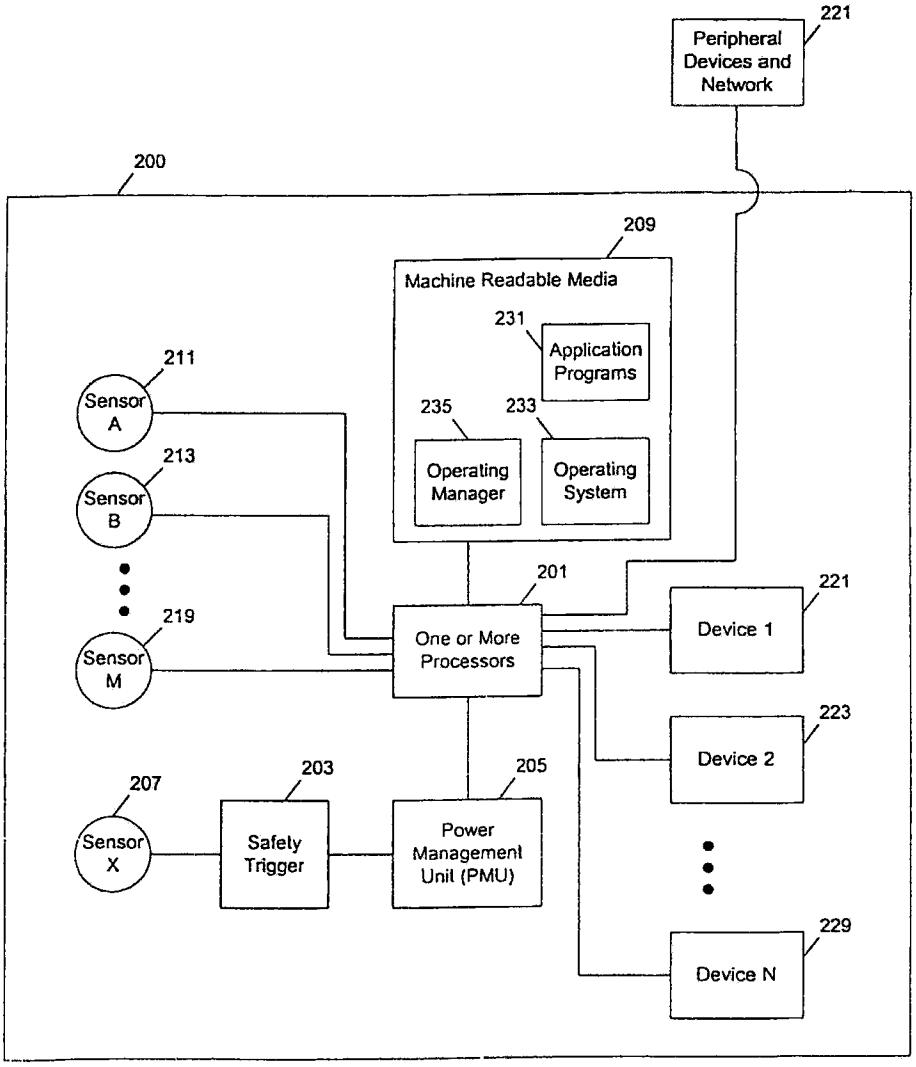

Peripheral Devices and Network
221
200
209
Machine Readable Media
231
Application Programs
235
233
Operating Manager
Operating System
211
Sensor A
213
Sensor B
219
Sensor M
201
One or More Processors
221
Device 1
223
Device 2
229
Device N
207
Sensor X
203
Safety Trigger
205
Power Management Unit (PMU)

Fig. 2

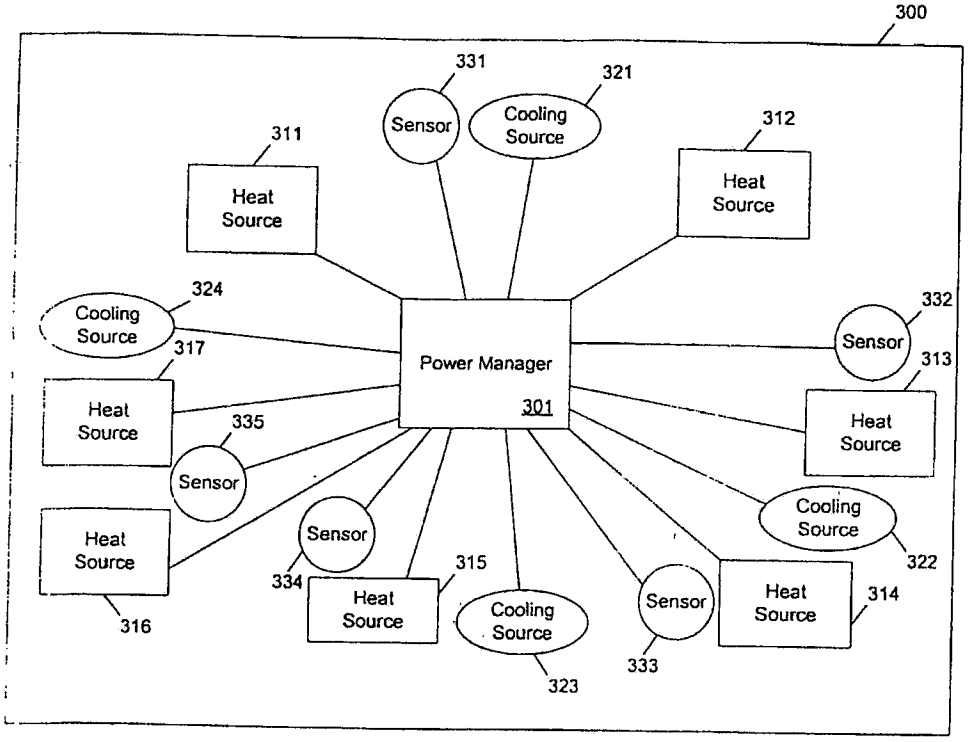

300
331
321
Sensor
Cooling Source
311
312
Heat Source
Heat Source
324
332
Cooling Source
Sensor
317
313
Power Manager
301
Heat Source
Heat Source
335
Sensor
Cooling Source
322
Sensor
Heat Source
316
334
315
Heat Source
Cooling Source
Sensor
Heat Source
314
323
333

Fig. 3

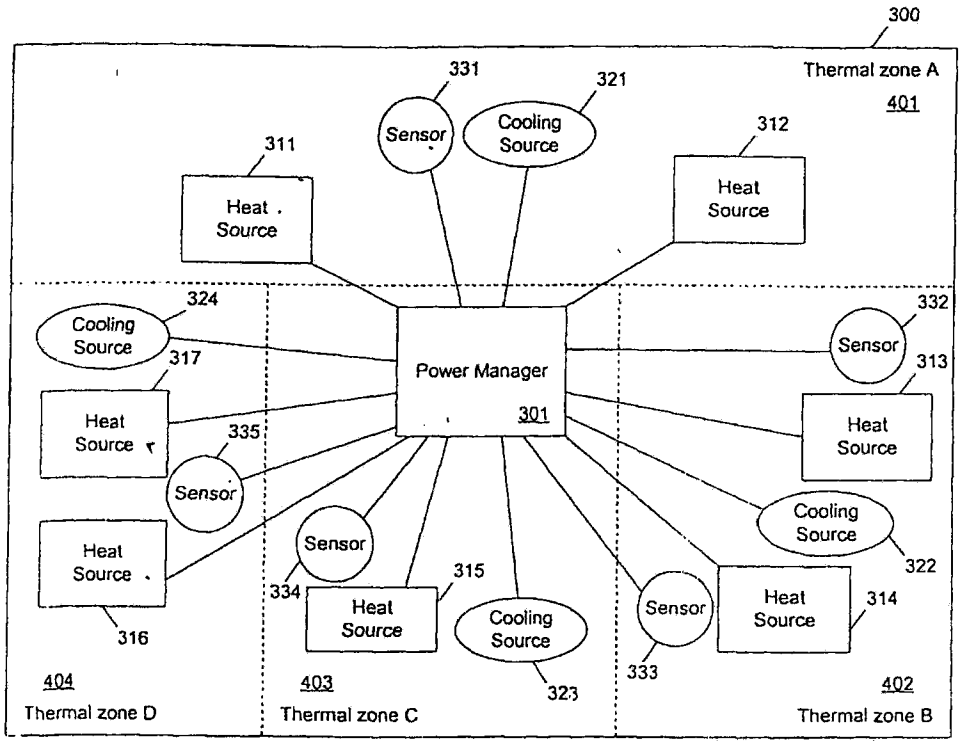

300
Thermal zone A
401
331
Sensor
321
Cooling Source
311
Heat Source
312
Heat Source
324
Cooling Source
317
Heat Source
335
Sensor
Heat Source
316
Power Manager
301
332
Sensor
313
Heat Source
Cooling Source
322
Sensor
334
Heat Source
315
Cooling Source
323
Sensor
333
Heat Source
314
404
Thermal zone D
403
Thermal zone C
402
Thermal zone B

Fig. 4

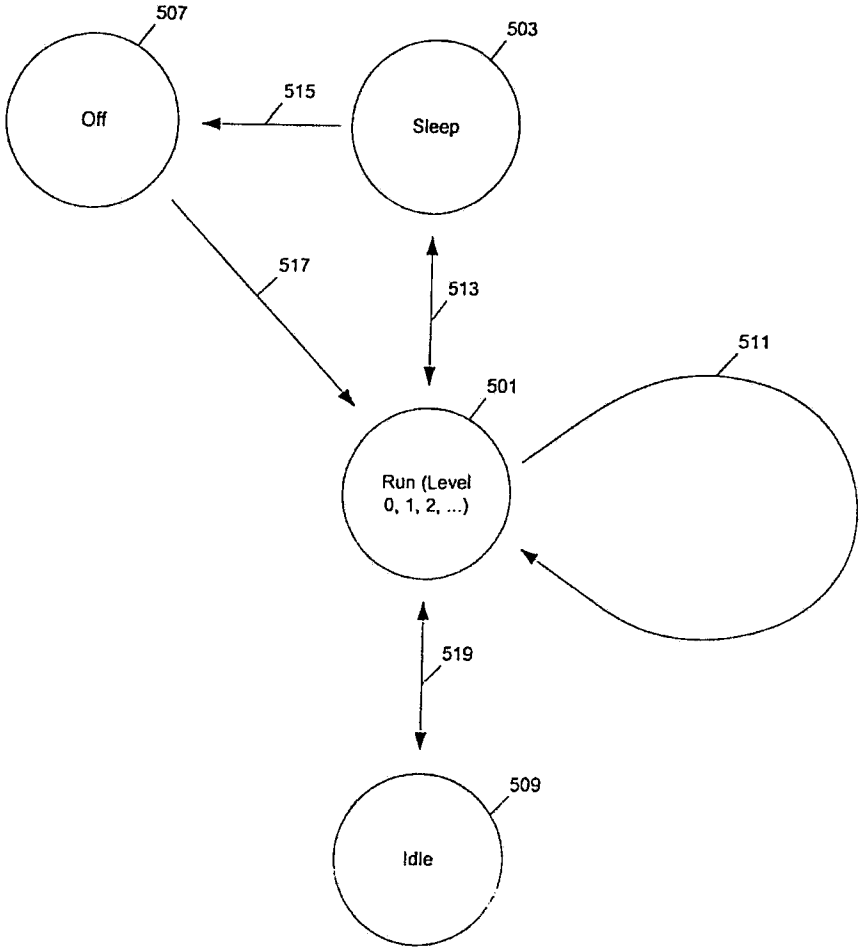

507
503
Off
515
Sleep
517
513
511
501
Run (Level
0, 1, 2, ...)
519
509
Idle

Fig. 5

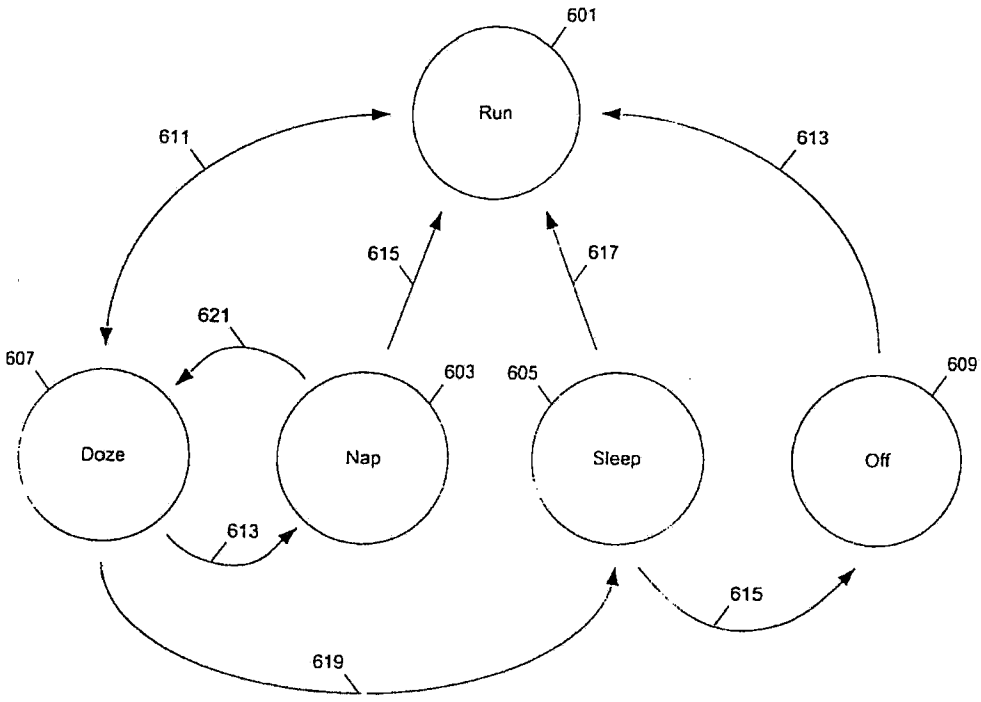

601
Run
611
613
615
617
621
607
Doze
603
Nap
605
Sleep
609
Off
613
615
619

Fig. 6

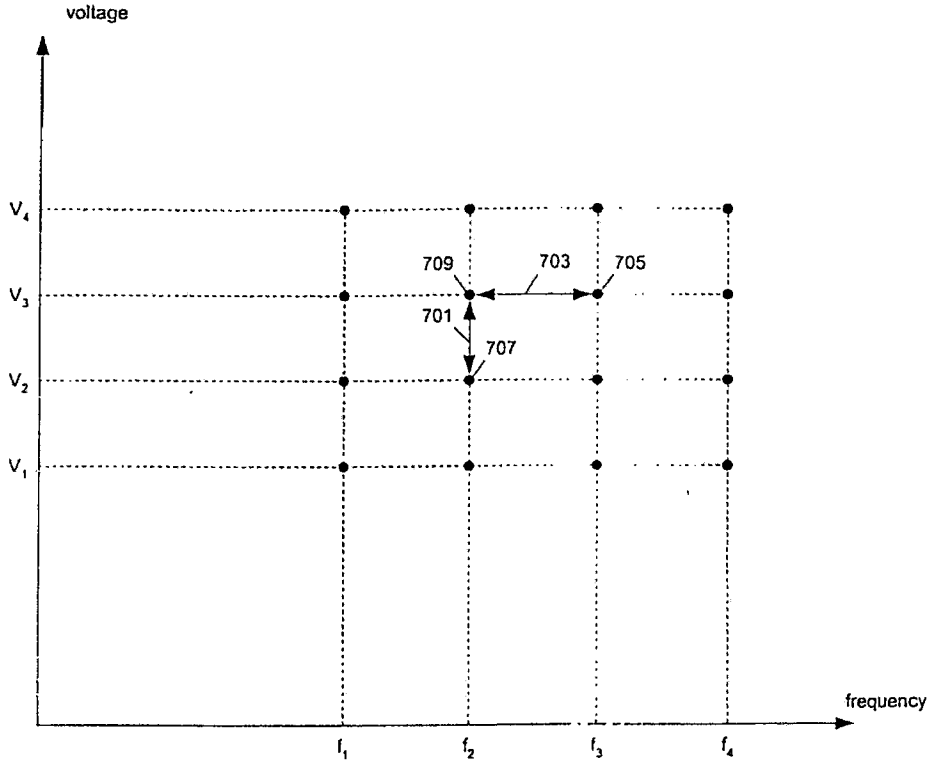


Fig. 7

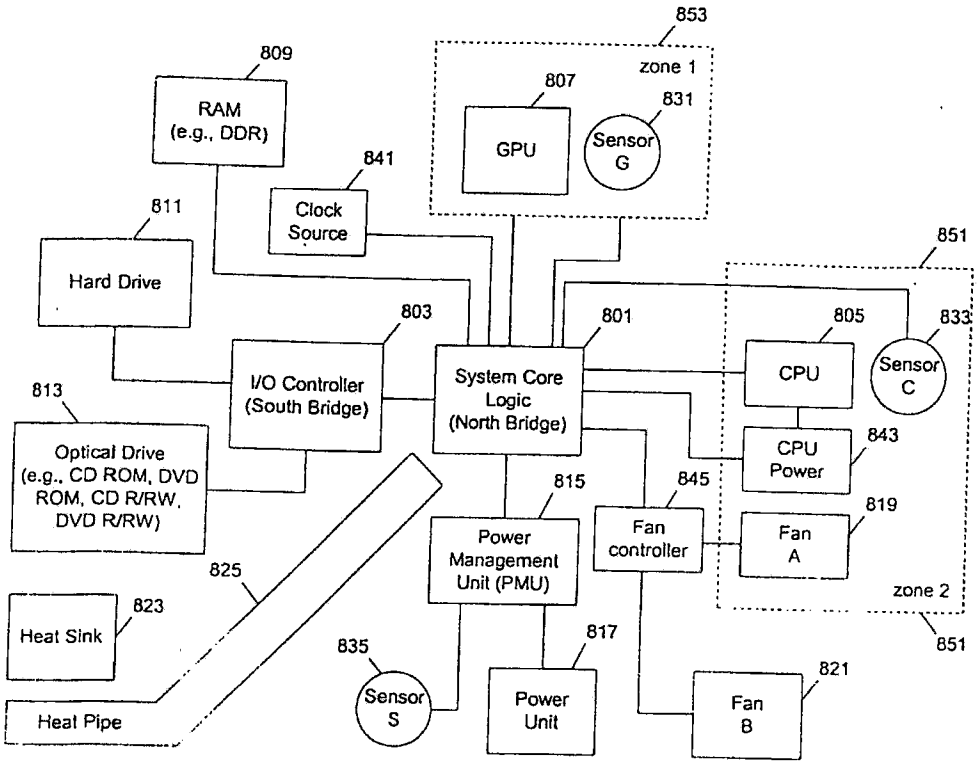

853
zone 1
809
RAM
(e.g., DDR)
807
GPU
831
Sensor
G
841
Clock
Source
811
Hard Drive
851
803
801
805
833
I/O Controller
(South Bridge)
System Core
Logic
(North Bridge)
CPU
Sensor
C
813
Optical Drive
(e.g., CD ROM, DVD
ROM, CD R/RW,
DVD R/RW)
CPU
Power
843
815
845
819
Power
Management
Unit (PMU)
Fan
controller
Fan
A
825
zone 2
823
Heat Sink
851
835
817
821
Sensor
S
Power
Unit
Fan
B
Heat Pipe

Fig. 8

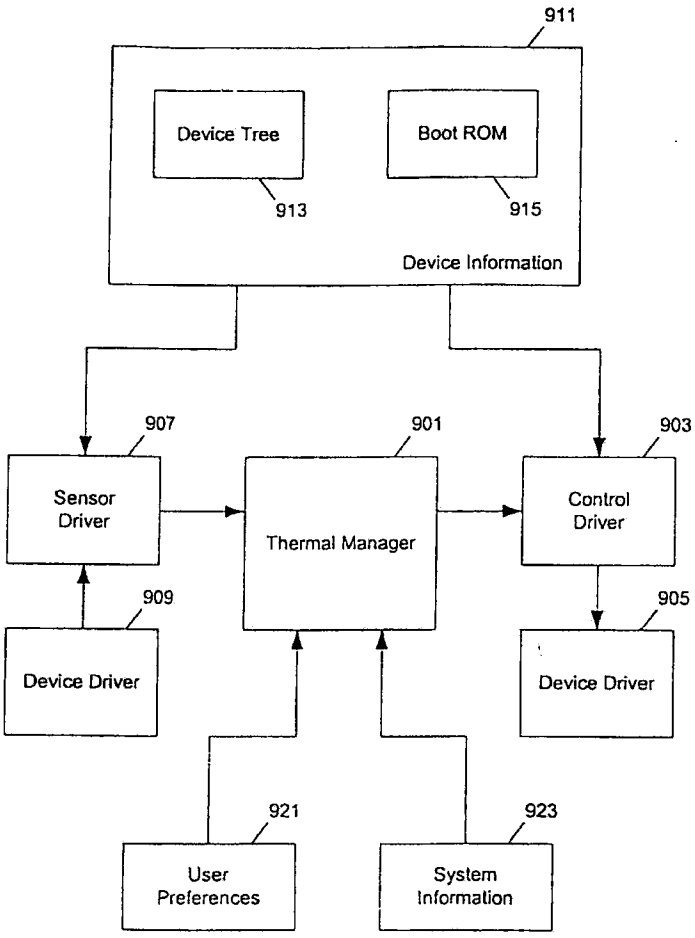

911
Device Tree
913
Boot ROM
915
Device Information
907
901
903
Sensor Driver
Thermal Manager
Control Driver
909
905
Device Driver
Device Driver
921
923
User Preferences
System Information

Fig. 9

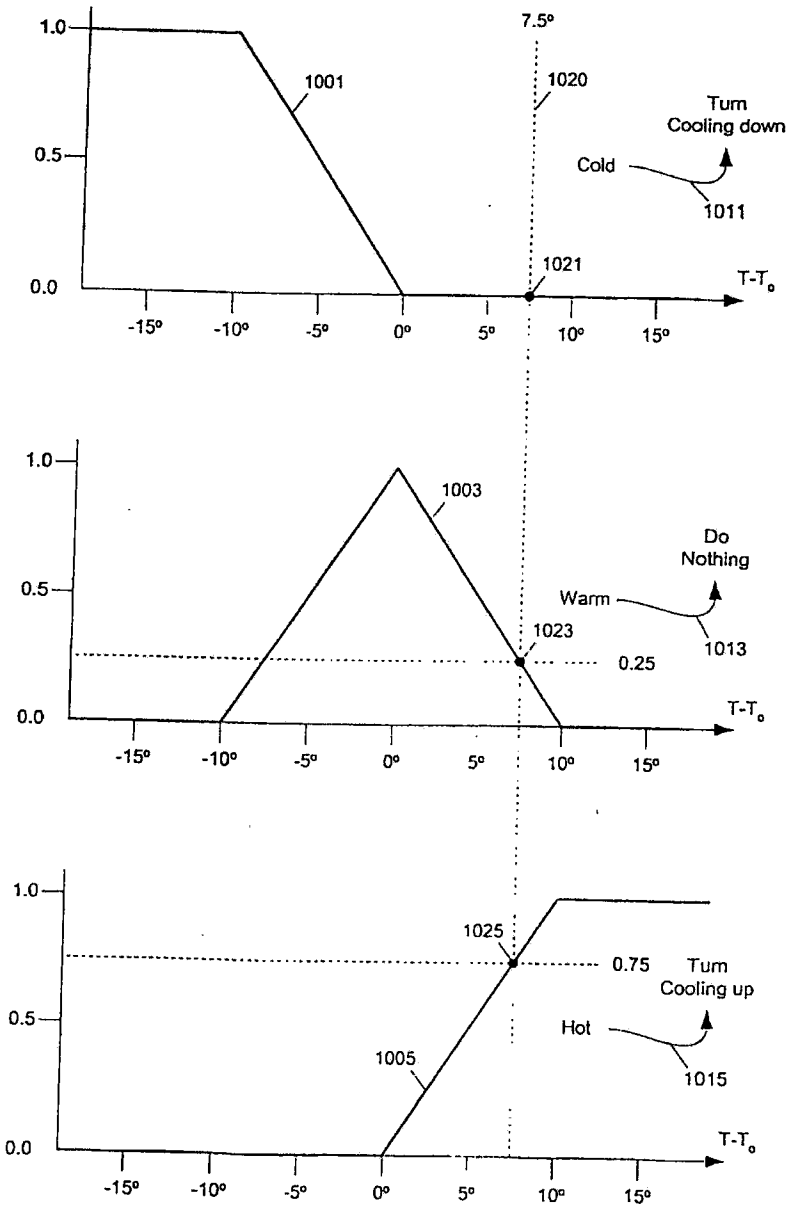

1.0
0.5
0.0
1001
7.5°
1020
Turn
Cooling down
Cold
1011
1021
T-T$_o$
-15°
-10°
-5°
0°
5°
10°
15°
1.0
0.5
0.0
1003
Do
Nothing
Warm
1013
1023
0.25
T-T$_o$
-15°
-10°
-5°
0°
5°
10°
15°
1.0
0.5
0.0
1025
0.75
Turn
Cooling up
Hot
1015
1005
T-T$_o$
-15°
-10°
-5°
0°
5°
10°
15°

Fig. 10

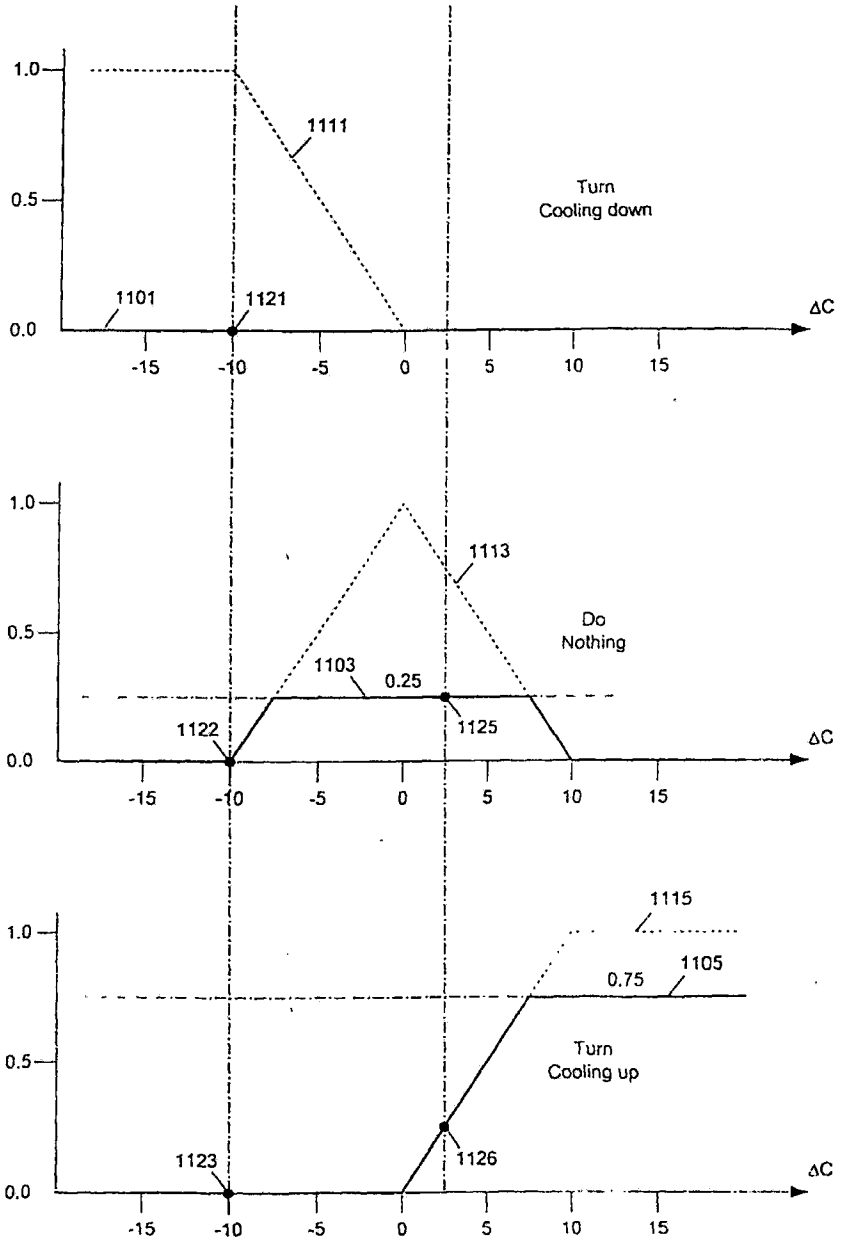

1.0
0.5
0.0
1111
Turn
Cooling down
1101
1121
ΔC
-15
-10
-5
0
5
10
15
1.0
0.5
0.0
1113
Do
Nothing
1103
0.25
1122
1125
ΔC
-15
-10
-5
0
5
10
15
1.0
0.5
0.0
1115
1105
0.75
Turn
Cooling up
1123
1126
ΔC
-15
-10
-5
0
5
10
15

Fig. 11

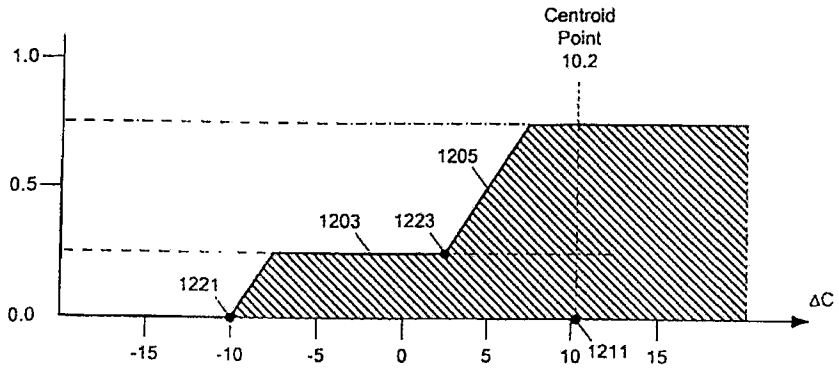

Centroid
Point
10.2
1.0
0.5
0.0
1205
1203
1223
1221
ΔC
-15
-10
-5
0
5
10
1211
15

Fig. 12

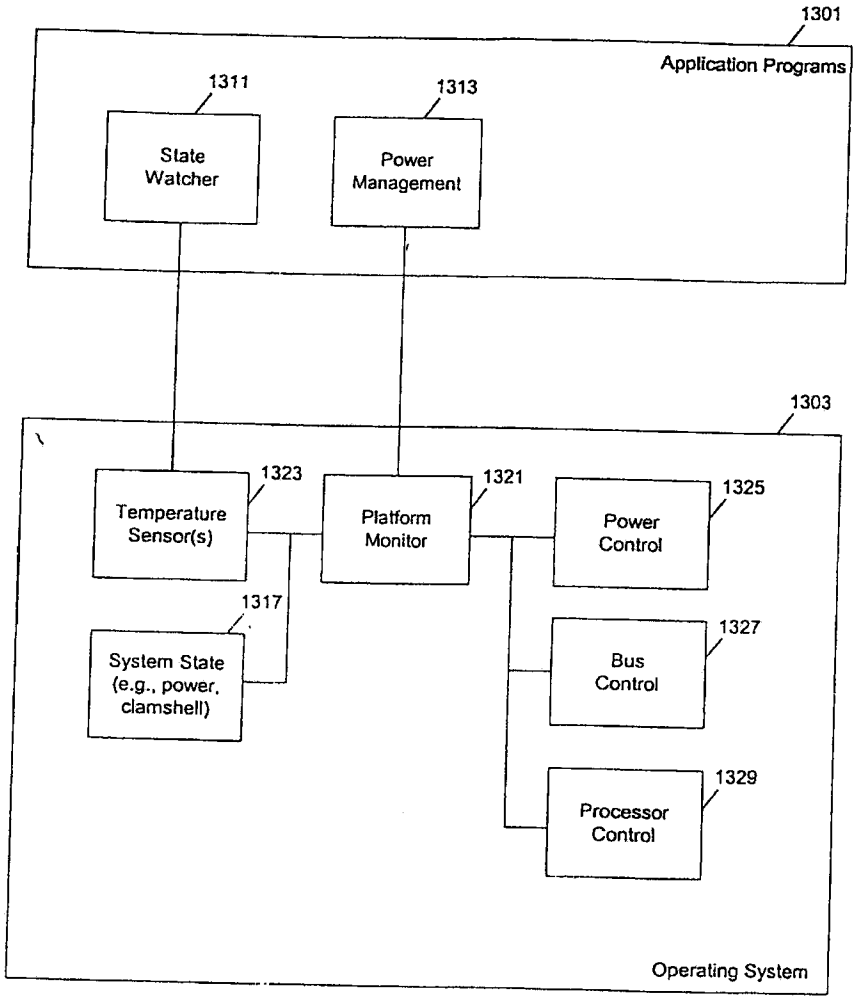

1301
Application Programs
1311
State Watcher
1313
Power Management
1303
1323
Temperature Sensor(s)
1321
Platform Monitor
1325
Power Control
1317
System State (e.g., power, clamshell)
1327
Bus Control
1329
Processor Control
Operating System

Fig. 13

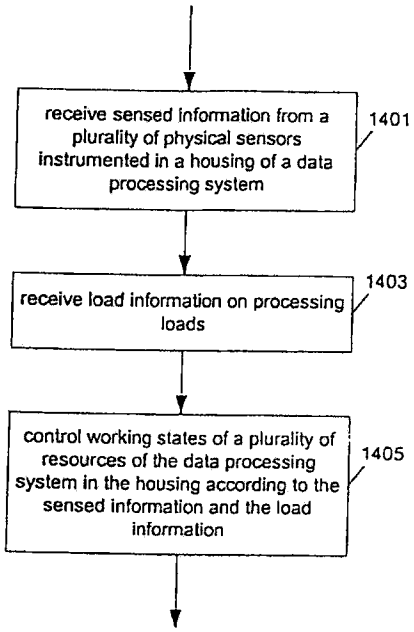

receive sensed information from a plurality of physical sensors instrumented in a housing of a data processing system
1401
receive load information on processing loads
1403
control working states of a plurality of resources of the data processing system in the housing according to the sensed information and the load information
1405

Fig. 14

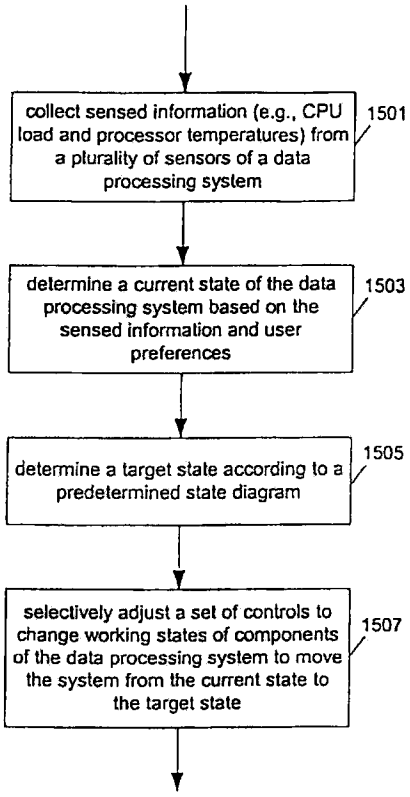

collect sensed information (e.g., CPU load and processor temperatures) from a plurality of sensors of a data processing system
1501
determine a current state of the data processing system based on the sensed information and user preferences
1503
determine a target state according to a predetermined state diagram
1505
selectively adjust a set of controls to change working states of components of the data processing system to move the system from the current state to the target state
1507

Fig. 15

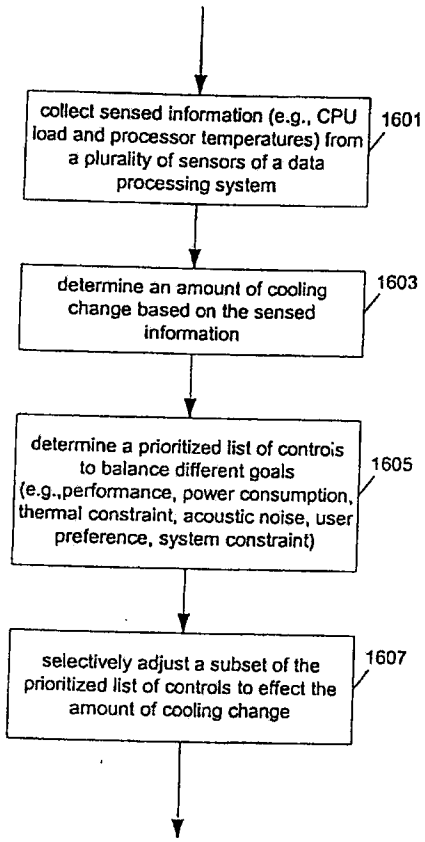

collect sensed information (e.g., CPU load and processor temperatures) from a plurality of sensors of a data processing system
1601
determine an amount of cooling change based on the sensed information
1603
determine a prioritized list of controls to balance different goals (e.g.,performance, power consumption, thermal constraint, acoustic noise, user preference, system constraint)
1605
selectively adjust a subset of the prioritized list of controls to effect the amount of cooling change
1607

Fig. 16

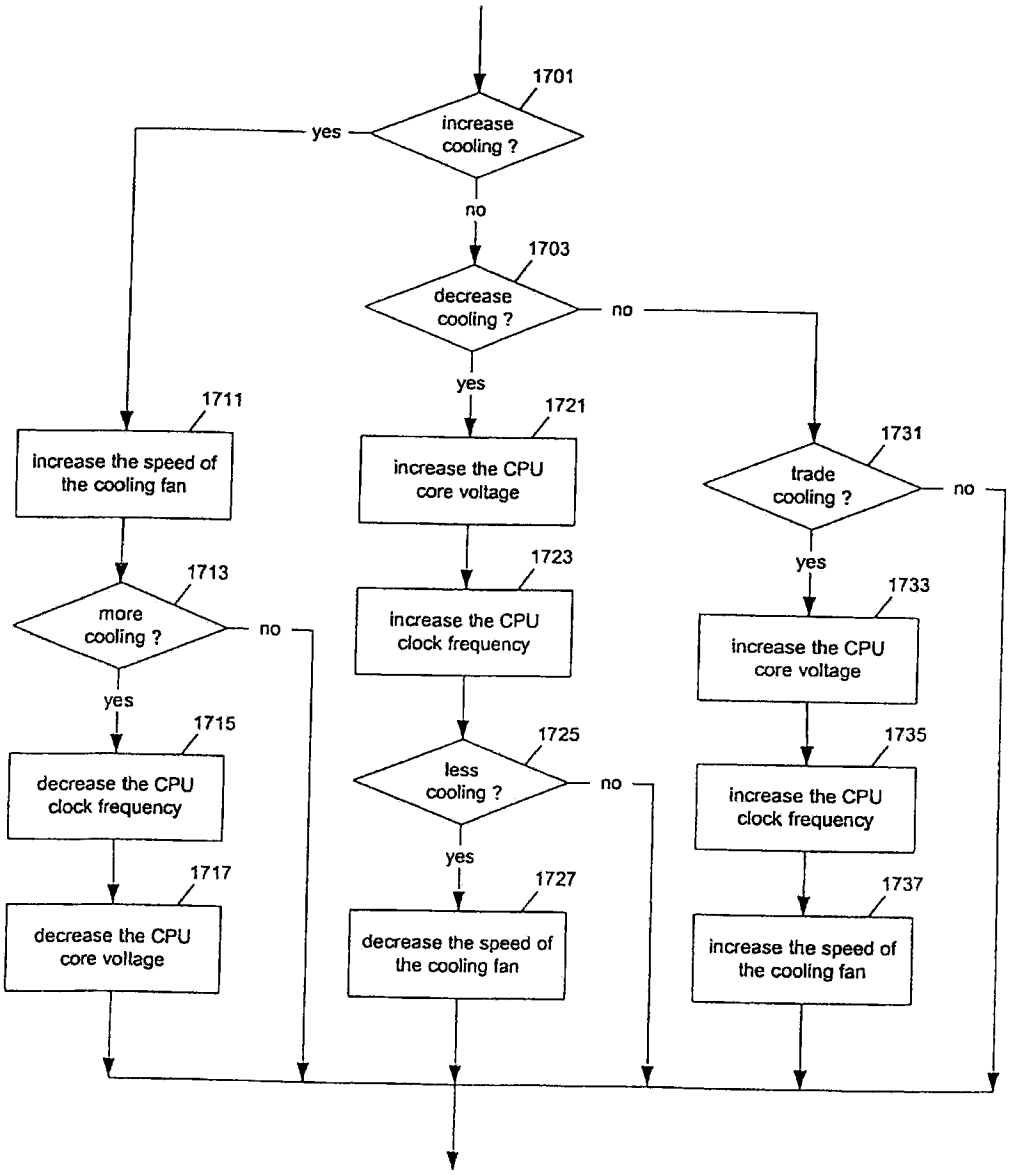

1701
increase cooling ?
yes
no
1703
decrease cooling ?
no
yes
1711
increase the speed of the cooling fan
1713
more cooling ?
no
yes
1715
decrease the CPU clock frequency
1717
decrease the CPU core voltage
1721
increase the CPU core voltage
1723
increase the CPU clock frequency
1725
less cooling ?
no
yes
1727
decrease the speed of the cooling fan
1731
trade cooling ?
no
yes
1733
increase the CPU core voltage
1735
increase the CPU clock frequency
1737
increase the speed of the cooling fan

Fig. 17

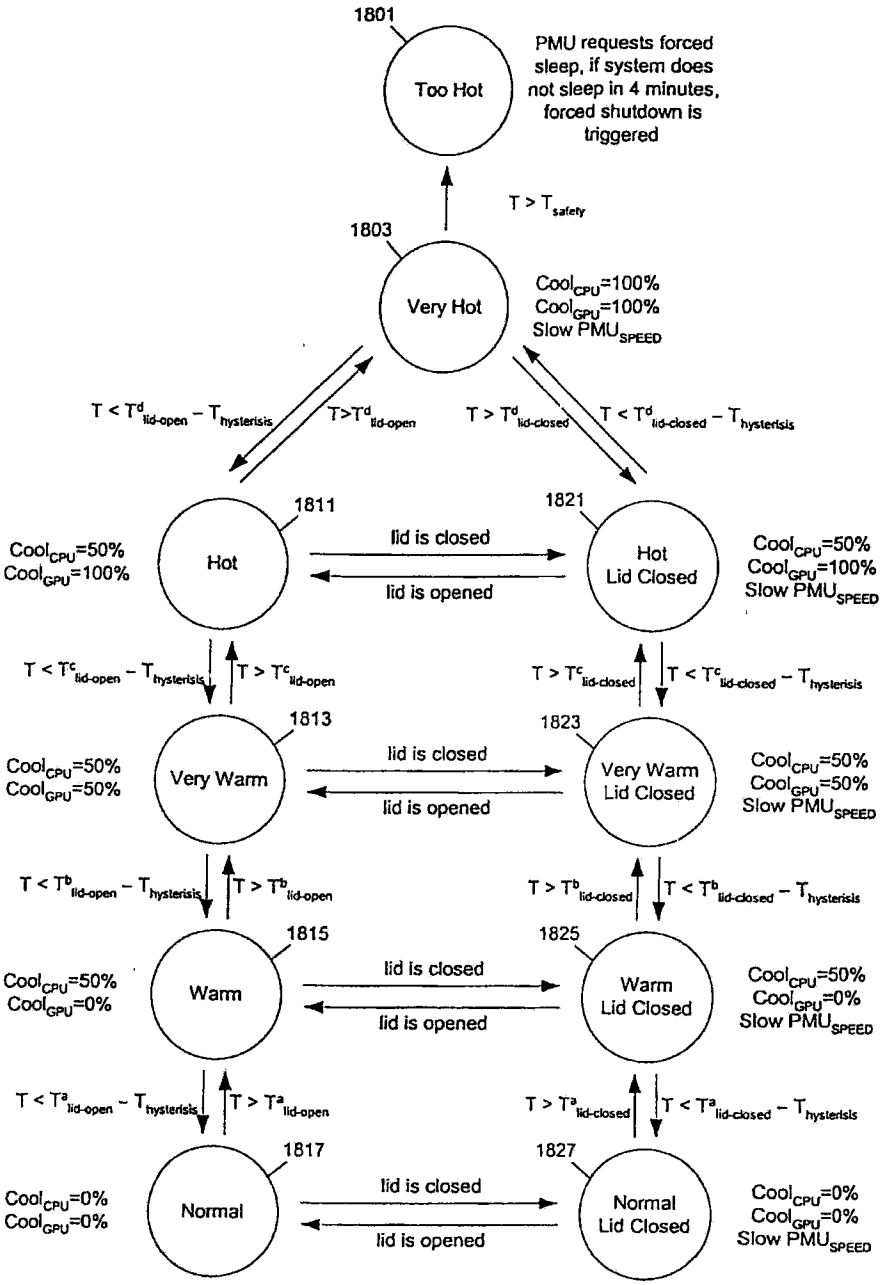

1801
Too Hot
PMU requests forced sleep, if system does not sleep in 4 minutes, forced shutdown is triggered
T > T safety
1803
Very Hot
Cool CPU =100%
Cool GPU =100%
Slow PMU SPEED
T < T d lid-open – T hysterisis
T>T d lid-open
T > T d lid-closed
T < T d lid-closed – T hysterisis
1811
Hot
Cool CPU =50%
Cool GPU =100%
lid is closed
lid is opened
1821
Hot Lid Closed
Cool CPU =50%
Cool GPU =100%
Slow PMU SPEED
T < T c lid-open – T hysterisis
T > T c lid-open
T > T c lid-closed
T < T c lid-closed – T hysterisis
1813
Very Warm
Cool CPU =50%
Cool GPU =50%
lid is closed
lid is opened
1823
Very Warm Lid Closed
Cool CPU =50%
Cool GPU =50%
Slow PMU SPEED
T < T b lid-open – T hysterisis
T > T b lid-open
T > T b lid-closed
T < T b lid-closed – T hysterisis
1815
Warm
Cool CPU =50%
Cool GPU =0%
lid is closed
lid is opened
1825
Warm Lid Closed
Cool CPU =50%
Cool GPU =0%
Slow PMU SPEED
T < T a lid-open – T hysterisis
T > T a lid-open
T > T a lid-closed
T < T a lid-closed – T hysterisis
1817
Normal
Cool CPU =0%
Cool GPU =0%
lid is closed
lid is opened
1827
Normal Lid Closed
Cool CPU =0%
Cool GPU =0%
Slow PMU SPEED

Fig. 18